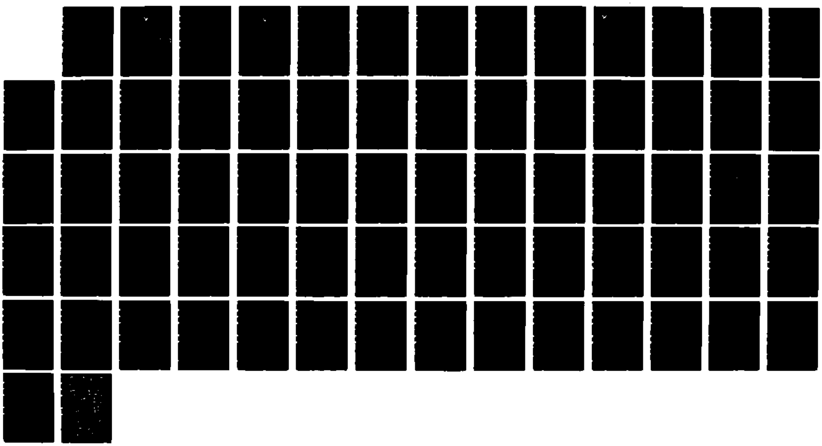
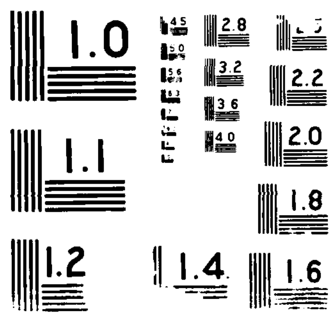
AD-A194 276 LINEAR GOAL PROGRAMMING AS A MILITARY DECISION AID(U) 1/1
AIR COMMAND AND STAFF COLL MAXWELL AFB AL J F POWELL
APR 88 ACSC-88-2155

UNCLASSIFIED F/G 12/4 NL




1.0
1.1
1.2
1.4
1.6
1.8
2.0
2.2
2.8
3.2
3.6
4.0
4.5

(2)

FILE COPY

AD-A194 276

AIR COMMAND AND STAFF COLLEGE

DTIC ELECTE JUN 0 7 1988 S D H

STUDENT REPORT

LINEAR GOAL PROGRAMMING AS A MILITARY DECISION AID

MAJOR JAMES F. POWELL 88-2155

"insights into tomorrow"

DISTRIBUTION STATEMENT A

Approved for public release; Distribution Unlimited

88

DISCLAIMER

The views and conclusions expressed in this document are those of the author. They are not intended and should not be thought to represent official ideas, attitudes, or policies of any agency of the United States Government. The author has not had special access to official information or ideas and has employed only open-source material available to any writer on this subject.

This document is the property of the United States Government. It is available for distribution to the general public. A loan copy of the document may be obtained from the Air University Interlibrary Loan Service (AUL/LDEX, Maxwell AFB, Alabama, 36112-5564) or the Defense Technical Information Center. Request must include the author's name and complete title of the study.

This document may be reproduced for use in other research reports or educational pursuits contingent upon the following stipulations:

- Reproduction rights do not extend to any copyrighted material that may be contained in the research report.

- All reproduced copies must contain the following credit line: "Reprinted by permission of the Air Command and Staff College."

- All reproduced copies must contain the name(s) of the report's author(s).

- If format modification is necessary to better serve the user's needs, adjustments may be made to this report--this authorization does not extend to copyrighted information or material. The following statement must accompany the modified document: "Adapted from Air Command and Staff College Research Report ____(number)____ entitled ____(title)____ ________________ by ______(author)."

- This notice must be included with any reproduced or adapted portions of this document.

REPORT NUMBER 88-2155

TITLE LINEAR GOAL PROGRAMMING AS A MILITARY DECISION AID

AUTHOR(S) MAJOR JAMES F. POWELL, USAF

FACULTY ADVISOR MAJOR JACK B. ROBBINS, ACSC/EDJ

SPONSOR COLONEL HALBERT R. SMART
OJCS/J-3
CHIEF, COMBAT OPERATIONS SUPPORT DIVISION

Submitted to the faculty in partial fulfillment of requirements for graduation.

AIR COMMAND AND STAFF COLLEGE
AIR UNIVERSITY
MAXWELL AFB, AL 36112-5542

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT STATEMENT "A" Approved for public release; Distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 88-2155	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION ACSC/EDC 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) MAXWELL AFB, AL 36112-5542	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

LINEAR GOAL PROGRAMMING AS A MILITARY DECISION AID

12. PERSONAL AUTHOR(S)

POWELL, JAMES F. MAJOR, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
	FROM ________ TO ______	1988 APRIL	64

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report explores the use of linear goal programming as a military decision aid. Linear programming, as well as linear goal programming, is explained with a basic graphic example and definitions. The complicated weapon allocation problem facing the combined arms commanders in todays battlefield is defined in linear programming terms, developed into computerized linear programming format, and solved on a Apple IIE home computer. This solution provides an optimum weapon selection for the given situation. Additional capabilities, applications, and future potential are also discussed. Copyrighted software and printout are included.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL ACSC/EDC MAXWELL AFB AL 36112-5542	22b. TELEPHONE (Include Area Code) (205) 293-2867 22c. OFFICE SYMBOL

PREFACE

This report explains how linear programming operates, highlights the differences between linear programming and linear goal programming, and develops a data set to apply linear goal programming as a decision aid for a combined arms commander in a tactical situation. These data and software are demonstrated in computing an optimum solution for a weapon selection problem. Additional information is presented on more powerful capabilities not demonstrated as well as additional proposed military applications.

Previous work on this subject was recently accomplished by the author in the University of Nebraska MBA program under the tutelage of Professor Marc Schneiderjans. Special thanks is due Dr. Schneiderjans for his patient instruction and generous copyright release.

ABOUT THE AUTHOR

Major James F. Powell is a Electronic Warfare Officer with a broad range of Air Force experience. His initial assignment, after navigator and electronic warfare officer training, was as a combat certified electronic warfare officer aboard B-52's. After four years in this aircraft Major Powell cross trained into RC-135's where he flew the Cobra Ball aircraft. In this assignment he was selected as a Reconnaissance Crew Commander directing the entire mission aboard this aircraft and was additionally selected as the Standardization/Evaluation Elint Branch Chief. In this capacity he was responsible for all tasking guidance for this national platform. Following this assignment Major Powell was selected to serve on the Operations staff at HQ SAC, Offutt AFB NE. In this capacity he initially served as the B-1B Electronic Combat Support Branch Chief. Here he was responsible for all electronic combat support for the B-1B and spearheaded the development of several computer programs to provide this support. Subsequently, Major Powell was selected as the Special Technical Operations Division Chief. In this regard he served as the focal point for all HQ SAC Special Technical Operations initiatives. Major Powell is a 1973 graduate of the University of Houston with a BS degree in Electrical and Electronics Technology. He earned his commission in 1974 through OTS and subsequently earned an MBA from the University of Nebraska in 1987.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

EXECUTIVE SUMMARY

REPORT NUMBER 88-2155

AUTHOR(S) MAJOR JAMES F. POWELL

TITLE LINEAR GOAL PROGRAMMING AS A MILITARY DECISION AID

I. Purpose: To demonstrate the validity and utility of linear goal programming as a military decision aid tool.

II. Problem: The complexity of the modern battlefield, coupled with the speed, lethality, and vast range of weapon systems has surpassed the ability of the current joint planning staffs to insure optimum allocation of all weapon systems. Due to this complexity, the combined arms commander currently has no responsive method of ensuring optimum selection of weapons for application against various targets.

III. Method: Initially the report explains the basic operation of linear programming and how an optimum solution if derived. Additionally, representative, notional data sets for various weapon systems available to a combined arms commander are developed. Computerized linear goal programming processing of these data is demonstrated to provide an optimum weapon system selection.

IV. Conclusion: The type information required for linear programming to be applied is either available or can be derived. Computerized linear goal programming, operating on these data, offers a high degree of utility in the demonstrated combined arms case as well as other military applications.

V. Recommendation: HQ USAF conduct a detailed study of the utility of linear goal programming as a decision aid system.

Chapter One

THE COMBINED ARMS WEAPON SELECTION PROBLEM

The profession of military command has changed immensely throughout the years. This change is nowhere more apparent than in the weapons employed and the proliferation of targets in the conduct of warfare. In recent history the weapons available to the military commanders and the targets selected were relatively simple. The weapons consisted of hand held lances, bow and arrows, up to light artillery, while the targets consisted primarily of concentrations of enemy troops. The commander employing these weapons usually had firsthand knowledge of the application techniques and capabilities of the weapons he directed. Modern warfare, and its vast array of weapons and targets has changed this basic tenet of warfare.

Warfare has expanded to include weapons, even entire theaters of conflict, not dreamed of in most of recorded history. This, combined with the geographical expansion of theaters of warfare, the relative speed of war fighting, the range of weapons, and the proliferation of targets of all types, has exponentially increased the complexity of modern warfare. The historically recent additions of submarine warfare, air warfare, advanced armour warfare, the potential for space warfare, and many other advances have expanded the breadth of weapons employed to the point that no combined arms commander can be expected to have detailed knowledge of all the resources available to wage war.

Recognizing this eventuality, modern armies have adopted and expanded to include a staff organization to provide this basic knowledge of resources available so that these diverse systems might be properly employed in battle. These staff organizations often take the form of sizeable numbers of staff officers deployed with the headquarters elements to provide planning expertise for the weapons systems employed. This approach has worked well in ensuring that most weapons are employed properly but it has not allowed an overall capability to efficiently allocate all available weapon systems in relation to their individual strengths and weaknesses. In essence, there is no "big picture" plan of weapon allocation other than the mental evaluation of the commanders. The weapons involved have become so numerous and diverse in characteristics that this is rapidly becoming an impossible task.

Throughout history, these type calculations have made the difference between victory and defeat. Drawing from Sun Tzu, "Military tactics are like unto water; for water in its natural course runs away from high places and hastens downward. So in war, the way is to avoid what is strong and to strike at what is weak. Water shapes its course according to the nature of the ground over which it flows; the soldier works out his victory in relation to the foe he is facing."(2:29) Applied to today's environment, this axiom might can be construed to say the application of forces must be finely tuned and tailored according to the enemy faced. To accomplish this there must be a method to ensure the overall optimization of all forces employed in the combined arms theaters of today.

This paper demonstrates an optimization technique, linear goal programming, which can provide the overall optimization guideline required and perform as a valuable, dynamic, decision aid for the combined arms commander. The paper will present the basics of linear programming but will concentrate on the application of computer based linear goal programming techniques to a representative mix of weapon systems and targets to demonstrate the capabilities and adaptability of this approach.

Chapter Two

WHAT LINEAR PROGRAMMING IS AND HOW IT WORKS

LINEAR PROGRAMMING

This chapter will lay the groundwork for the entire following project. Since linear programming is possibly a new topic for many readers, the logical starting point is a common, layman's definition of linear programming. This step will ensure we all start with a common terminology and frame of mind. Building on this definition, we will then discuss the requirements for linear programming applications, the parts of a linear programming problem, how linear programming works, the limits of linear programming, and finally, how linear goal programming relates to linear programming.

Linear programming is best described as a mathematical technique used to find the one best, or optimum, solution for a given situation from a set of feasible solutions. Linear indicates that the relationships among the elements, or variables, can be expressed as proportional mathematical functions. Programming simply refers to the type model and its usage to "program" elements of the solution. Linear programming, as an optimization technique, began in 1947 with G. B. Dantzig's interactive process. (3:4) This technique has almost constantly been refined and grew to include linear goal programming beginning with a text written by A. Charnes and W. W. Cooper, Management Models and Industrial Applications of Linear Programming in 1961.(3:5) Since this time linear programming and linear goal programming have continued to gain acceptance as valuable management tools, and have been applied to many diverse management systems.

Linear programming is applicable to a wide range of management problems, however, there are four basic conditions which must exist before it can be considered the appropriate quantitative technique.(1:192-193)

1. The decision maker is attempting to achieve a specific objective. (objectives in the case of linear goal programming)
2. Alternative solutions are available. (several answers might "fit", but only one is optimum)
3. Resources are scarce.
4. The objective (objectives in linear goal programming) and resource limitations can be expressed as linear mathematical equations or inequalities.

If these conditions exist linear programming should be considered as a valid optimization technique and thought should be given to transforming the problem to a linear programming format.

In order to express a problem in linear programming form you must be familiar with the parts and terminology of a linear programming problem. The terminology presented here is that usually applied to computer formatted problems, and will be used throughout this project as it is geared to a computer derived solution. The following example illustrates the components of a linear programming minimization problem.

$$\text{Min: } Z = C_1 X_1 + C_2 X_2 + \ldots C_n X_n$$
$$\text{subject to:}$$
$$A_{11} X_1 + A_{12} X_2 + \ldots A_m X_n < b_1$$
$$A_{21} X_1 + A_{22} X_2 + \ldots A_{2n} X_n < b_2$$
$$A_{m1} X_1 + A_{m2} X_2 + \ldots A_{mn} X_N < b_m$$
$$\text{and: } X_1, X_2, \ldots X_n > 0$$

X = decision variables (the number of goods to be produced or resources allocated for the given solution).
C = contribution coefficient (how much each good or resource contributes to the given solution).
Z = unknown solved for (in a minimization problem it is usually an expression of combined resources required to provide the optimum solution).
b = side constraints (usually mathematical expressions of resource limitations, however may represent practically any limiting factors capable of being expressed as linear mathematical functions).

Theoretically, any problem is capable of being solved utilizing linear programming techniques if it can be expressed in these terms. It is often beneficial to think in graphic terms to better understand the processes and manipulations employed in linear programming to reach an optimum solution. The graphic solution presented will show the relationships of the elements and the manipulation required to reach an optimum solution. In this case the manipulation is done graphically, however utilizing the format presented above it is possible to convert this expression to a form readily adaptable to computer processing. This is what will be done with the stated problem facing the combined arms military commander later in this paper.

A GRAPHIC EXAMPLE OF HOW LINEAR PROGRAMMING WORKS

Note: This example is an adaptation of a problem from (7).

Problem: Consider two aircraft, the X1 and the X2. The X2 is slightly larger than the X1 and therefore can carry 5 bombs compared to 4 for the X1 (bombs of equal size). The X1, because it is faster requires 4000 lbs. of fuel per mission as compared to 2000 lbs. for the X2. The X2, because it is older, requires 6 hours maintenance preparation per mission as compared to 3 hours for the X1. Our squadron has 32000 lbs. of fuel and 36 maintenance hours available and is tasked with delivering the maximum bomb load for tomorrow's mission. How many of each aircraft should be utilized ?

Problem Restatement:

Bomb load; X1 = 4
X2 = 5

Aircraft	Fuel Required	Maintenance hrs. available
X1	4000	3
X2	2000	6
Total	32000	36

Linear programming Formulation:

Maximize Z (bomb load)
X1 = number of aircraft X1 to utilize| decision variables
X2 = number of aircraft X2 to utilize|
Maximize: Z = 4X1 + 5X2| objective function
subject to:
4000X1 + 2000X2 < 32000 (fuel avail.) | constraints
3X1 + 6X2 < 36 (maintenance hrs) |

The first step in the graphic solution is to solve for the overall limits for each constraint. This is done by setting each decision variable equal to zero, solving for the remaining decision variable, and then graphing the resulting line.

Assume: 4000X1 + 2000X1 = 32000
let: X1 = 0
then: 4000(0) + 2000X2 = 32000
2000X2 = 32000
X2 = 16
let: X2 = 0
then: 4000X1 + 2000(0) = 32000
4000X1 = 32000
X1 = 8

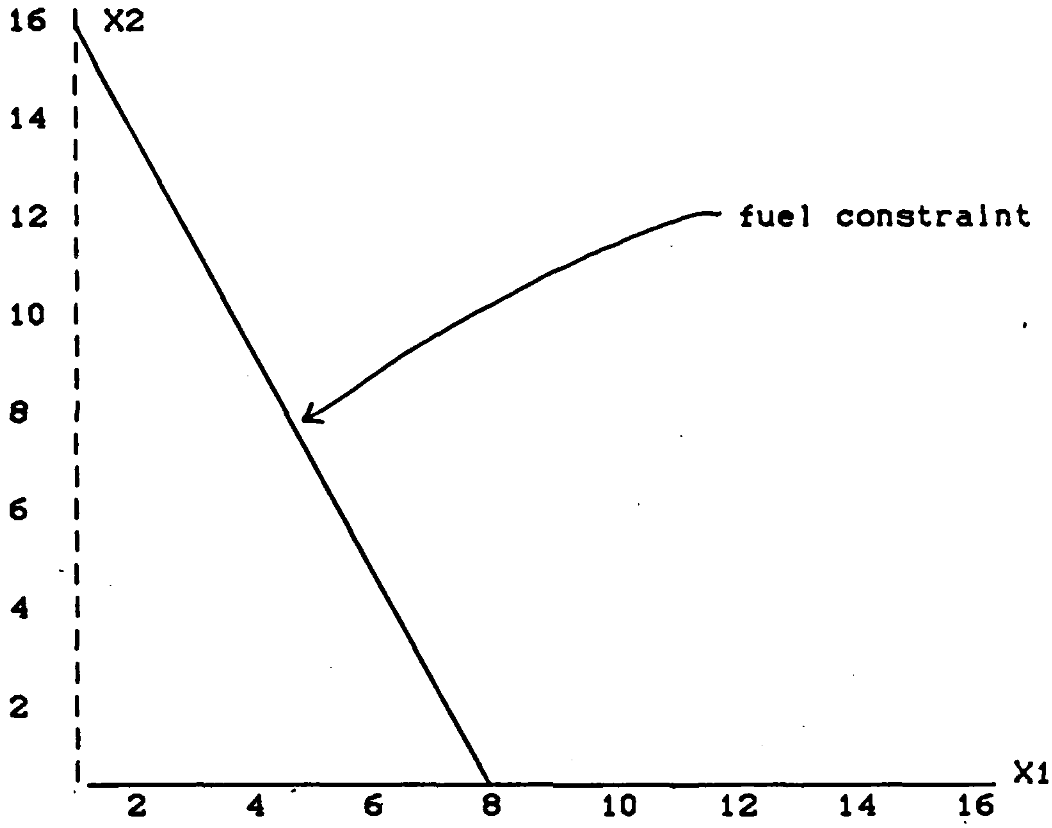
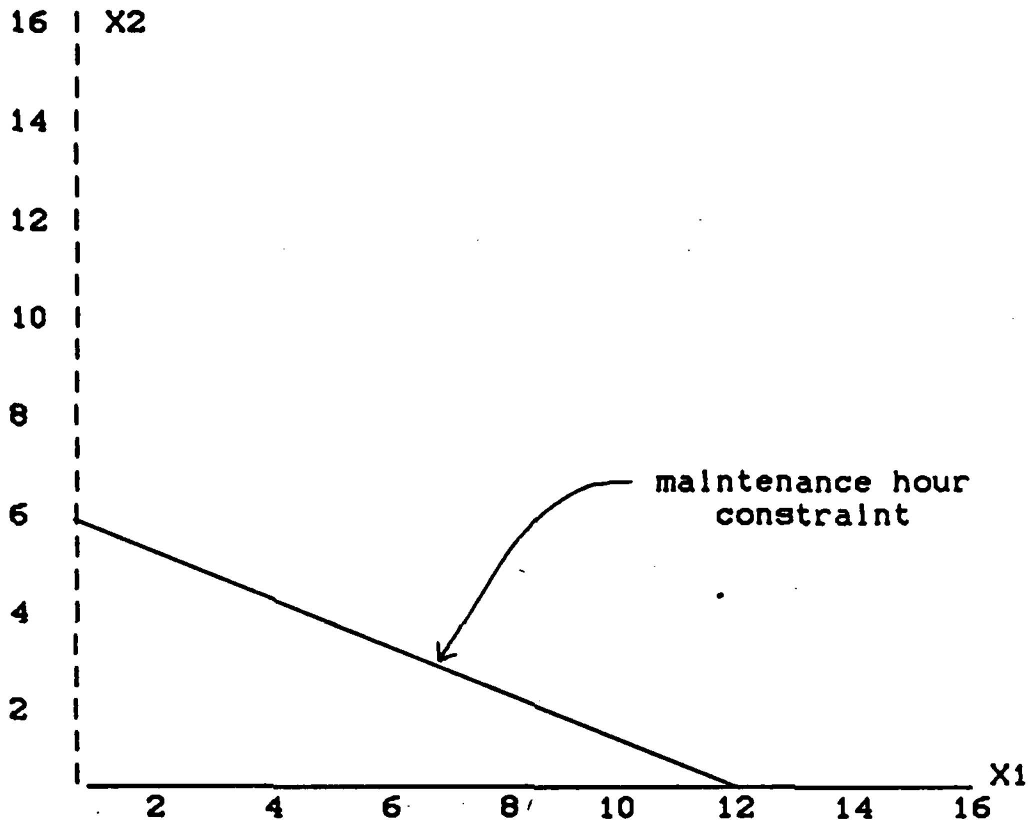


FIGURE 1

This process is then repeated for the remaining decision constraint.

Assume: $3X1 + 6X2 = 36$
let: $X1 = 0$
then: $3(0) + 6X2 = 36$
$6X2 = 36$
$X2 = 6$
let: $X2 = 0$
then: $3X1 + 6(0) = 36$
$3X1 = 36$
$X1 = 12$



16
14
12
10
8
6
4
2
X2
maintenance hour constraint
2
4
6
8
10
12
14
16
X1

FIGURE 2

The next step is to graph the area of feasible solutions. This area is defined as that area to the left of both constraint lines when they are combined on a graph. Theoretically, any point within this area is a feasible solution.

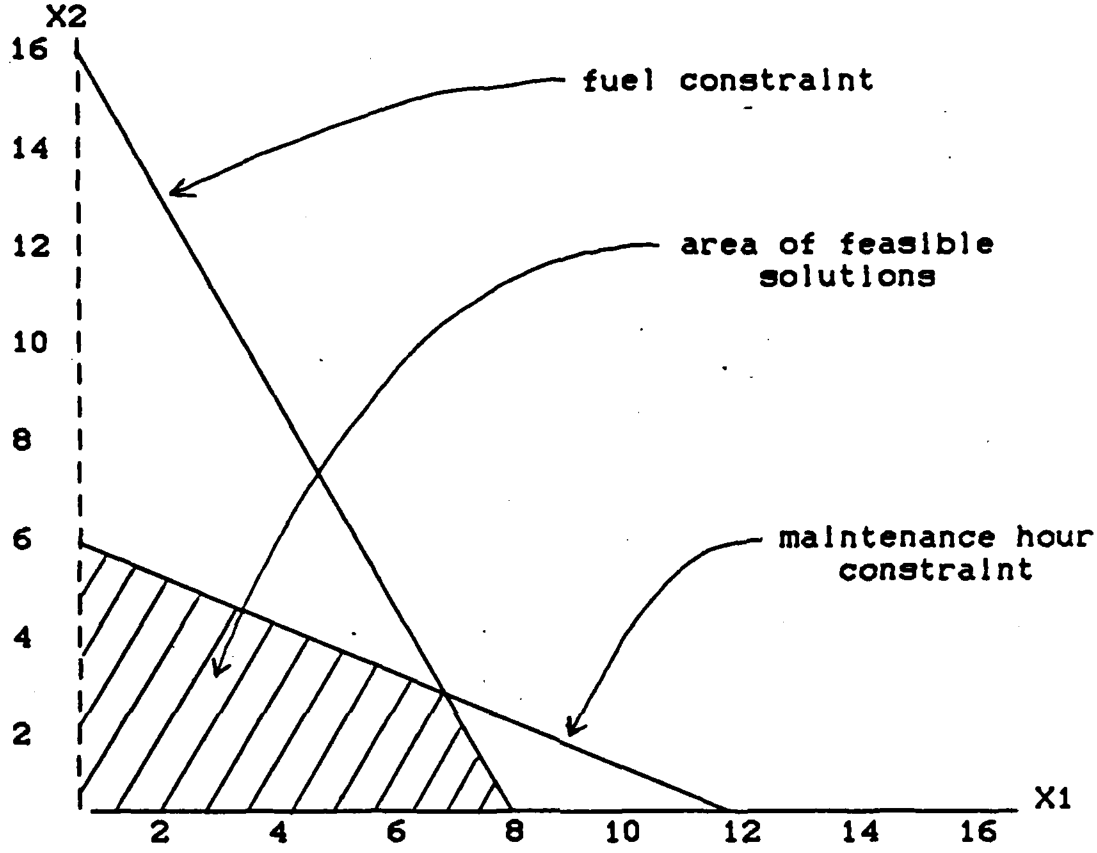


FIGURE 3

The next step is to determine the slope of a feasible solution. In this step we pick a convenient number for Z, solve the objective function, and graph the result.

Let Z = 20

Then: $4X1 + 5X2 = 20$

let: $X1 = 0$	let: $X2 = 0$
then: $4(0) + 5X2 = 20$	then: $4X1 + 5(0) = 20$
$5X2 = 20$	$4X1 = 20$
$X2 = 4$	$X1 = 5$

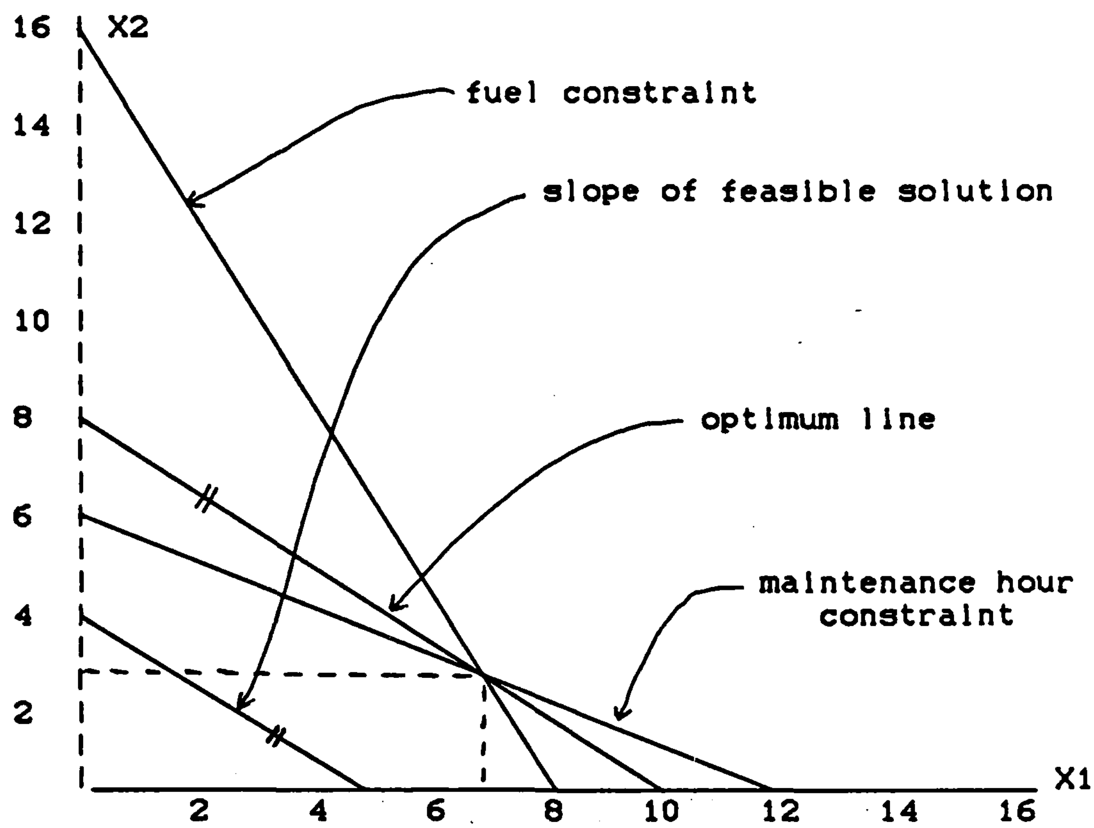


FIGURE 4

The final step is to draw a line parallel to this line at the last point of tangency to the area of feasible solutions (see figure 4, optimum line). Lines perpendicular to the axis from this point of tangency will intercept the axis at the optimum values for the decision variables.

$Z = 40$
$X1 = 6\ 2/3$
$X2 = 2\ 2/3$

In the real world we know we cannot fly 2/3 of an aircraft. Consider though if you were planning a mission of hundreds of aircraft. This solution is valid as long as the linear relationships are maintained, that is hours of maintenance, fuel, and bombs per aircraft remain the same. Then the solution may be expanded linearly or simply resolved. This example illustrates graphically the same process that occurs when we solve linear programming problems on a computer, the major difference being the computer does the "number crunching".

The same type solution will work for a minimization solution as well as a maximization solution. In the minimization case, the area of feasible solutions shifts to the right of the two constraint lines instead of to the left. Many computer programs are coded to handle either type solution. Computers additionally have the capabilities of handling large numbers of constraints as well as numerous decision variables thusly expanding the solution well beyond the two dimensional graphic solution capability. These capabilities will be demonstrated in the computer solution of the stated problem from chapter one.

Before we start this process it is important to recognize the limits of linear programming. As discussed to this point, linear programming is capable of considering and solving for only one objective. Recognizing this shortcoming, Y. Ijiri in 1965 published Management Goals and Accounting for Control which described the use of preemptive priority factors to allow the modeling of multiple conflicting objectives in accordance with their ranked importance in the objective function. Simply put, this new technique allowed for the simultaneous solution for multiple goals in priority order. This is the fundamental difference between linear programming and linear goal programming. Other differences include the capability to attach priority weights to specific constraints to dictate which are considered first. These differences set linear goal programming apart as a powerful, more manipulable form of linear programming. To take advantage of these advances we will utilize linear goal programming to solve the combined arms weapon allocation problem.

Chapter Three

PROBLEM STATEMENT IN LINEAR PROGRAMMING TERMS

DEFINITION PARAMETERS

Recall from chapter one, the problem to be addressed in this paper is the optimization of the various weapon systems available to the combined arms commander. This portion of the project will deal with the development and detail of the elements necessary to manipulate the stated problem with linear goal programming techniques.

To provide an orderly flow to this development process a representative group of weapon systems will be defined and representative constraints developed. Note at this point that the actual values employed in this example are notional values to serve for demonstration purposes only. The relative values employed are a result of the authors experience and on occasion are referenced to sources such as Fast Track. The use of these notional values versus precise, validated values is employed to preclude any classification issues. Additionally, the aim of this project is to demonstrate the utility of linear goal programming as a technique, therefore, precise values are not required.

The limitations of the micro software used in this demonstration must be considered prior to element construction. This software (appendix B) is constructed to operate on the Apple II plus, IIE, and IIC family of home computers and is limited due to the limited capabilities of these systems. These limits, thirty-five goal constraints, ten decision variables, and nine priorities must be kept in mind while defining the demonstration components.(3:115) It would be impractical to structure a problem with more elements than the demonstration software is capable of handling, and would not contribute to the validity of the demonstration. Additional software and processing capabilities are available to handle these larger applications. (3:201)

DECISION VARIABLES DEFINED

With these limits in mind, the definition of weapon systems available, or decision variables is the necessary first step in the development of demonstration exercise elements. In this example, weapon systems available will include the following systems.

X1 - tactical aircraft (conventional arms)
X2 - strategic aircraft (conventional arms)
X3 - tactical missiles (conventional arms)
X4 - strategic missiles (nuclear arms)
X5 - ground assault force (infantry and armor, conventional arms)
X6 - chemical munitions
X7 - tactical aircraft (smart conventional munitions)
X8 - unconventional warfare assault (special forces, etc.)
X9 - tactical missiles (nuclear arms)
X10 - strategic aircraft (nuclear arms)

Referencing the example presented in chapter two, these weapon systems will represent decision variables, or X values in the objective function. It is important to recall that the decision variables for an actual problem could be defined to any required degree of accuracy. As an example, they could be defined as different weapon loads on the same type aircraft. The wide range of decision variables presented here were selected to demonstrate the overall flexibility of the linear goal programming technique as applied to the combined arms problem.

CONSTRAINT DEFINITION

The second part of the construction process will focus on the definition of constraints. These constraints can be thought of as the overall problem set facing the combined arms commander. For example, he may desire to destroy a target but hesitates to use his most effective weapon system because of the overriding fear of escalation. In this case preventing escalation is his highest priority while target destruction assumes a lower priority. In this demonstration, escalation, target destruction, and other decision factors will be defined and modeled as constraints. The following factors will be modeled in this demonstration.

b1 - Timeliness: the relative time interval from execution to weapon arrival at target.

b2 - Probat lity of detection: the relative probability that the weapon system will be detected, identified, and countered prior to arrival at target.

b3 - Probability of target destruction: the relative probability of target destruction after weapon arrival.

b4 - Escalation factor: relative likelihood that use of this weapon system will lead to escalation of the present scenario.

b5 - Probability of personnel loss: relative probability that allied personnel will be lost during the application of the various weapon systems.

b6 - Weapon system availability: the constraint established by the gross number of each weapon system available.

Recalling again the example presented in chapter two, these decision factors will be modeled as constraints, or b, in this example. These constraints, like the decision variables, could be made as finite as desired in an actual application. The macro software at appendix A is capable of handling up to one hundred and fifty constraints and ten priorities providing a greatly increased capability to tailor an overall optimization model. (3:201)

Chapter Four

DATA DEVELOPMENT, MANIPULATION, AND PROCESSING

DATA DEVELOPMENT

This section of the report highlights one of the major problems facing the Department of Defense today. This is the problem of obtaining and utilizing accurate data when modeling systems from more than one command or service. Many of the services operating the weapons systems utilized in this model have well developed models and data for many of the constraints modeled here, but are hesitant to release this data outside the command or service. This hesitancy stems from the fear that once released, the data will be manipulated and used against the service or command in the PPBS cycle. We must overcome this hesitancy to provide the accurate, timely data to support modeling systems which reach across several organizations within the Department of Defense.

This chapter centers on further development of the linear goal programming elements defined in chapter three and culminates in their entry into a linear goal programming computer program. In order to manipulate the decision variables and constraints defined in chapter three with linear goal programming computer software it is necessary to assign numerical values to them. This section of the report will focus on the further development of these elements to allow their manipulation with a linear goal programming computer program.(appendix B) Numerical values will be assigned to each constraint as they are associated with a particular weapon system or decision variable. These values are simply the relative values associated with each weapon system for each constraint. The constraint relative value information is presented utilizing the prefixes assigned to each element in chapter three.

Timeliness is simply the relative time from execution to weapon arrival on target. Timeliness is minimum for the missiles in the example (they take the least time to reach the target) and is maximum for the ground forces. The required times are rated, the minimum equaling one, and the maximum equaling ten, with the intermediate values linearly expressed as values between one and ten in this example.

b1 - Timeliness: maximum value 10 (least timely)
minimum value 1 (most timely)

X1 - 3	X6 - 2
X2 - 5	X7 - 3
X3 - 1	X8 - 1
X4 - 2	X9 - 1
X5 - 9	X10 - 5

Probability of detection is the relative probability that a weapon system will be detected and countered prior to reaching the target. Detection models already exist for most of the systems in this model. In the case of the strategic systems, the ROPES (Route Penetration Evaluation System) operated at Headquarters Strategic Air Command (SAC) provides probability of detection based on the type threats expected to be encountered. This value is expressed as a percentage value representing the actual probability of detection. The actual numeric values and algorithms employed in this model are classified and the values are not released outside the headquarters. This case is representative of many of the other systems employed in this model. Probability of detection models exist, but the data is classified or is not approved for release outside the command.

The values used here are representative values based on Fast Sick , a Tactical Air Forces Employment Feasibility Exercise, values and various other unclassified publications. The values were derived by reviewing the threat level expected from our prime potential adversary (the Soviet Union), to be faced by each weapon system. Those systems whose detection is assured (ground assault forces) were given a maximum value of approximately ten (9). Those systems whose detection is very unlikely (tactical missiles) were given minimum values approaching one. The other systems with intermediate probabilities of detection are linearly represented on the scale between two and nine.

b2 - Probability of detection:
maximum value 10 (detection assured)
minimum value 1 (detection unlikely)

X1 - 3	X6 - 5
X2 - 4	X7 - 3
X3 - 2	X8 - 2
X4 - 6	X9 - 2
X5 - 9	X10 - 4

Probability of target destruction is simply the probability that the weapon system will destroy the intended target after arrival. The probabilities expressed here are derived from Fast Stick, Army FM 100-5, and common sense. Those systems with the highest probability of target destruction (nuclear weapons) are awarded the minimum value of one. Those systems less likely to destroy the target are awarded linearly higher values ranging from one to ten. Keep in mind that these values assume arrival at the target. Inability to reach the target is modeled in probability of detection.

b3 - Probability of target destruction:
maximum value 10 (destruction questionable)
minimum value 1 (destruction assured)

X1 - 4	X6 - 3
X2 - 3	X7 - 3
X3 - 4	X8 - 4
X4 - 1	X9 - 1
X5 - 2	X10 - 1

Probability of escalation is the most subjective constraint in this demonstration. As used here it is meant to be a military/political judgement of what the escalation potential of each weapon system is. In this example it is especially pertinent to the nuclear weapons.

Those systems with a high escalation potential (nuclear weapons) are given maximum values of nine while those systems with no escalation potential are given values of one. The intermediate systems are given linear values between one and nine representing their potential for escalating a given situation.

b4 - Probability of escalation:
maximum value 10 (escalation likely)
minimum value 1 (escalation unlikely)

X1 - 1	X6 - 6
X2 - 3	X7 - 2
X3 - 2	X8 - 2
X4 - 9	X9 - 8
X5 - 1	X10 - 9

Probability of personnel loss is a restated constraint containing two major factors, the probability of detection combined with the number of personnel exposed to hostile fire to execute each weapon system. Those systems with a low probability of personnel loss (strategic and tactical missiles) are given minimum values. Those systems employing large numbers of personnel and a high probability of detection (ground assault force) are given maximum values. The systems employing fewer people and having a lower probability of detection are given linear values between one and nine.

b5 - Probability of personnel loss:
maximum value 10 (loss likely)
minimum value 1 (loss unlikely)

X1 - 3	X6 - 1
X2 - 3	X7 - 3
X3 - 1	X8 - 8
X4 - 1	X9 - 1
X5 - 9	X10 - 3

Weapon system availability is the only constraint that is not presented in the form of relative values. In this instance, the constraint values are structured to represent the numbers of actual weapon systems available for each decision variable. These constraints are necessary to preclude the modeling of more resources than what might actually be available.

b6 - Weapon system availability

X1 - 100	X6 - 2
X2 - 2	X7 - 8
X3 - 5	X8 - 5
X4 - 10	X9 - 2
X5 - 5	X10 - 2

DATA MANIPULATION

Following the attachment of numerical values demonstrated above, it becomes necessary to arrange the constraints into a matrix format. This step is necessary to allow the developed constraints to later be translated directly to linear goal programming format. The constraint matrix for the constraints developed above follows. In order to limit clutter in the matrix, only the previously assigned prefixes are used to denote the constraints and decision variables.

Constraint Matrix

Constraint	b1	b2	b3	b4	b5	b6
Variable						
X1	3	3	4	9	3	100
X2	5	4	3	7	3	2
X3	1	2	4	8	1	5
X4	2	6	1	1	1	10
X5	9	9	2	9	9	5
X6	2	5	3	4	1	2
X7	3	3	3	8	3	8
X8	1	2	4	8	8	5
X9	1	2	1	2	1	2
X10	5	4	1	1	3	2

The next sequential step in translating the already developed constraints and decision variable into a computer understandable form requires that they be expressed in linear goal programming format. This step is a direct extension from the constraint format detailed above. Simply stated, the functions are summed, by columns to indicate the total contribution of all decision variables for each constraint.

3X1 + 5X2 + X3 + 2X4 + 9X5 + 2X6 + 3X7 + X8 + X9 + 5X10 = Timeliness

3X1 + 4X2 + 2X3 + 6X4 + 9X5 + 5X6 + 3X7 + 2X8 + 2X9 + 4X10 = Probability of detection

4X1 + 3X2 + 4X3 + X4 + 2X5 + 3X6 + 3X7 + 4X8 + X9 + X10 = Probability of target destruction

X1 + 3X2 + 2X3 + 9X4 + X5 + 6X6 + 2X7 + 2X8 + 8X9 + 9X10 = Probability of escalation

3X1 + 3X2 + X3 + X4 + 9X5 + X6 + 3X7 + 8X8 + X9 + 3X10 = Probability of personnel loss

Constraint b6, actual number of resources available is treated differently. As previously mentioned, these numbers represent the actual numbers of resources available instead of relative contribution values. Since they are individual resource values, they are required to be expressed as individual constraints applicable only to the decision variable they limit.
X1 = 100 (conventionally armed tactical aircraft available)
X2 = 2 (conventionally armed strategic aircraft available
X3 = 5 (conventional tactical missiles available)
X4 = 10 (nuclear strategic missiles available)
X5 = 5 (ground assault forces available)
X6 = 2 (chemical munitions available)
X7 = 8 (smart munitions armed tactical aircraft available)
X8 = 5 (unconventional warfare assault forces available)
X9 = 2 (nuclear armed tactical missiles available)
X10 = 2 (nuclear armed strategic aircraft available)

The next step in the development process requires some prior discussion of the actual software to be employed as well as linear goal programming software in general. The method of coding for most linear goal programming software operates against the deviation factor for each constraint. For example, if a minimization problem were being run the system would function to minimize positive deviation thusly forcing the affected variables to a value less than the maximum limit. Additionally, if priorities are assigned, the process is structured to compute each constraint in the order of the priority assigned to it, priority one being first and so on to the last priority. This allows the linear goal programming software to optimize a solution based on several constraints in priority order.

In order to allow for this capability it is necessary to prioritize the constraints previously developed. The problem, as stated, is developed as a linear goal programming minimization problem. As such, the software to be employed will operate to minimize the use of resources (defined as decision variables in the problem) while minimizing the collateral constraints in priority order. To provide this capability in the demonstration problem the following notional priorities will be assigned;

1. Minimize probability of personnel loss
2. Minimize probability of detection
3. Minimize probability of escalation.

These priorities are for demonstration only. The selection of real priorities is the prime area where the combined arms commander can tailor the linear goal programming technique to fit his existing tactical or strategic situation. The commander can select priorities to fit the existing situation and easily adapt these choices to the situation as it changes. This adaptability is one of the key benefits of linear goal programming. It allows it to be rapidly adjusted to the dynamic situation.

The final step in developing the stated problem in linear goal programming format is the writing of the objective statement. The objective statement is a summation of all the constraints, in priority order, into one minimization statement. It may be compared to the overall statement of objectives. In the case of the example, it will be a minimization statement (minimize scarce resources) and will define the minimization of each constraint in priority order by minimizing the positive deviation associated with each constraint. For the developed example the objective function is as follows:

Min: $Z = P1d5 + P2d2 + P3d4 + d1 + d3 + d6 + d7 + d8 + d9 + d10 + d11 + d12 + d13 + d14 + d15$

COMPUTER DATA PROCESSING

The statement above is the final manipulation required for input into the linear goal programming software to be used for this demonstration. (appendix B) It is beyond the scope of this paper to give a detailed explanation of the software and the operations it performs to obtain an optimum solution. The basic functioning of the software performs an operation similar to the graphic solution presented in chapter two only with additional dimensions and allowing for priorities.

With this in mind, the problem, as developed and defined is entered into the linear goal programming software on an Apple IIE home computer. Directions for this operation, as well as an example, can be found in Linear Goal Programming, page 122-125. (3:122-125)

The results of this processing demonstration are included as Appendix C. The resultant computer printout is in three essential parts. The first part (all that area above coefficients in tableau) is a display of the inputed weights and priorities. This area serves as a record to compare and check the output values printed below.

The second portion, coefficients in tableau, is a printout of the actual computer manipulation process. The computer utilizes a tableau solution process as an alternative to the graphic process to compute the optimum solution. Since the graphic process, illustrated earlier, is limited to simple solutions and graphic depictions it is not suited to computer manipulation. The tableau process employed consists of arranging the developed information into numerical tableau format and then manipulating these tableaus to provide an optimum solution. The tableau process is very much like the graphic process only it is done strictly with numerical manipulation to take advantage of the computer capabilities.

The third and last portion of the printout is the solution variable and goal display. This portion of the printout displays the values for the optimum solution with the given input values. In other words, with the previously developed values and stated priorities these values represent the optimum numbers of weapons systems to employ. The only relevant values are those with decision variable prefixes, X3 and X8, in the left column and positive values in the right column. The optimum solution for this problem is to employ 5 conventionally armed tactical missiles and 5 unconventional warfare assault forces. The second portion of this area displays any unachieved goals. In this case we had only three stated priority goals: minimize personnel loss, detection probability, and escalation potential. The displayed value for P4 is present because all constraints other than those stated above are entered as fourth priority goals and this is a composite value for them. Since they are not priority goals for the purposes of this demonstration this value is superfluous.

With this problem, once all the values have been inputed, the Apple computer required approximately four minutes to provide the attached solution. The four minutes time included approximately two and one half minutes actual computing time and approximately one and one half minutes printing time.

Chapter Five

SOFTWARE LIMITS AND ADDITIONAL CAPABILITIES

SOFTWARE LIMITS

As is evident by this point the demonstrated software is severely limited. The previously mentioned limits of thirty-five goal constraints, ten decision variables, and nine priorities limit this software to only the simplest goal programming applications (3:201). This limit is evident in the rough groupings of constraints required in the demonstration problem. With a more powerful program it would be possible to greatly refine the constraints to provide a much more realistic representation. With more powerful software, constraints like probability of target destruction, might be broken down to provide probabilities for several different types of targets and the decision variables, like tactical aircraft, might be expanded to provide variables for each distinct model of tactical aircraft available. These expansions would provide a more realistic, refined and usable product.

The demonstration software is additionally limited in the method employed to print out the optimum solution. Without a significant level of prior knowledge of the software it is difficult to determine what the separate areas and lines of the printout represent. With a more powerful, better refined software package it would be possible to provide a printout easily readable without any prior knowledge.

ADDITIONAL CAPABILITIES

The shortcoming of this software are solved on linear goal programming software hosted on larger computers. The macro computer software included as attachment A has a much greater manipulation capability than that demonstrated. This software allows for 150 goal constraints, 150 decision variables, and 10 preemptive priority levels (3:114). These expanded capabilities would allow many of the expansions to constraints and variables mentioned above to provide a more refined product. This macro software provides a tenfold increase in variable definition capability and a three fold increase in constraint definition capability when compared to the demonstrated micro level software.

In addition to the expanded definition capability, the macro level software provides a much more understandable printout. It provides the values of the decision variables as well as an analysis of the deviations and priority accomplishments. These analyses allow a much better presentation of the optimum solution values and the processing required to compute them.

Chapter Six

OBSERVATIONS AND RECOMMENDATIONS

OBSERVATIONS

This paper has demonstrated linear goal programming as a military decision aid system, however, linear programming has many other diverse uses, many applicable to the military. As a normative model linear goal programming may be applied to problems to minimize transportation costs, assign personnel to projects, and any type of problem dealing with the allocation of scarce resources (4:633). These type applications can provide utility to military maintenance organizations, supply organizations, or any other type military organization dealing with the allocation of any resource, including our most valuable resource, people. These applications, with the guaranteed optimum solution could save the military significant money as well as manpower hours in operating expenses.

The civilian business establishment has already recognized the power and potential of linear programming. In a recent survey (1984) almost three of every five respondent firms reported using linear programming in the production management area (1:193). Although not strictly in the production business, linear programming can be applied to many of the similar resource applications problems in the military.

Linear programming, in a different form is widely applied in electronics development and design. The basic Boolean algebra and Karnaugh map applications in electronic design led the way to the development of linear programming as a business tool (5:32). These efforts were initially developed to minimize electronic circuits in the design process and later led to the further development of linear programming as a business cost minimization tool. From this point they have developed as total decision aid systems applicable to a wide range of business and military applications.

The military services are presently making only limited use of linear programming. As presently employed, linear programming is used for some direct weapon allocation problems, limited parts control problems, and several other low order applications. In all of these cases linear programming is locally employed and does not cross command or service lines of responsibility. Also, to the authors knowledge, linear goal programming is yet to be applied in the military environment.

RECOMMENDATION

The next step in the development of linear programming and linear goal programming should be their adoption by the military as standard decision aid tools. With the proper software development linear programming can provide utility to the joint forces commander, the maintenance officer, the supply organization, numerous other military organizations, and it can ultimately provide the rapid decision potential required in today's battlefield. This rapid decision potential is additionally applicable to the battlefield of tomorrow and might provide the a key input in any system requiring a quick, optimum decision when several alternatives are available.

BIBLIOGRAPHY

A. REFERENCES CITED

Books

1. Boone, Louis E. and David L. Kurtz. Principles of Management. New York, NY: Random House Inc., 1984.

2. Clavell, James, Ed.. The Art of War by Sun Tzu. New York, NY: Delacorte Press, 1983.

3. Schniederjans, Marc J. Linear Goal Programming. Princeton, New Jersey: Petrocelli Books, 1984.

4. Szilagyi, Andrew D. Jr.. Management and Performance. Glenview, Illinois: Scott, Foresman and Company, 1984.

5. Wickes, William E.. Logic Design with Integrated Circuits. New York, NY: John Wiley and Sons, Inc., 1968.

Official Documents

6. Air University, Air Command and Staff College. Fast Stick, A Tactical Air Forces Employment Feasibility Exercise. Maxwell AFB, Alabama: 1986.

Other Sources

7. Powell, James F. Class notes from GBA-915, Decision Sciences, University of Nebraska, College of Business: Omaha, Nebraska, 1987.

B. RELATED SOURCES

Books

Adam, Everett E. and Ronald J. Ebert. Production and Operations Management. Englewood cliffs, New Jersey: Prentice-Hall, Inc., 1982.

Breckenbach, Edwin F., Irving Drooyan and William Wooton. College Algebra. Belmont, California: Wadsworth Publishing Company Inc., 1968.

Digman, Lester A. Strategic Management. Plano, Texas: Business Publications, Inc., 1986.

Ledley, Robert Steven. Digital Computer and Control Engineering. New York, NY: McGraw-Hill Book Company, Inc.,1960.

Maurice, S. Charles and Charles W. Smithson. Managerial Economics. Homewood, Illinois: Richard D. Irwin, Inc., 1985.

Articles and Periodicals

Dixon, W. L., J. M. Mellichamp and S. L. Mitchell. "Ballistic Missile Defense Technology Management with Goal Programming." Interfaces, Vol. 10, No. 5 (Oct 80), pp. 68-75.

Fitzsimmons, J. A. and R. S. Sullivan. "Goal Programming - Model for Readiness and Optimal Redeployment of Forces." Socio - Economic Planning Sciences, Vol. 12, No. 5 (1978), pp. 215-220.

Greenwood, A. G., B. W. Taylor and A. J. Keown. "An Integer Goal Programming Model for Determining Military Aircraft Expenditures." Journal of the Operational Research Society, Vol. 34, No. 5 (May 1983), pp. 379-390.

Nussbaum, D. A.. "Goal Programming as an Aid to Resource Management." Defense Systems Management Review, Vol. 3, No. 2, (Spring 80), pp. 28-33.

APPENDICES

29

Appendix A

LGP Macro Computer Program

```
C
C          THIS PROGRAM IS A DUAL SIMPLEX GOAL PROGRAMMING
C             ALGORITHM.
C          IT HAS BEEN DIMENSIONED FOR 150 DECISION
C             VARIABLES, 150 CONSTRAINTS, AND 10 GOAL PRIORITY
              LEVELS.

           IMPLICIT REAL*8(A-H,P-Z)
           INTEGER*2 ITIME(15)
           COMMON NROW,NCOL,NVAR,NPRT,KTEST,ITER
           COMMON /R1/ BASIS(150,300)
           COMMON /R2/ VALC(11,300),VALB(11,150)
           COMMON /R3/ PRHS(150),RHS(150)
           COMMON /I1/ IBASIC(150),JCOL(300)
           CALL TIMDAT (ITIME, INTS(15))
           WRITE(6,888) ITIME(4),ITIME(5),ITIME(6)
       888 FORMAT(' MIN',I7,5X,'SEC',I4,5X,'TICKS',I5)
           WRITE(6,889) ITIME(7),ITIME(8),ITIME(9),ITIME(10)
       889 FORMAT (' CPU S',I4,3X,'CPU T',I5,5X,'IO S',I4,3X,'IO T',I5)
           CALL START
           CALL SIMPLX
           CALL FINISH
           CALL TIMDAT (ITIME,INTS(15))
           WRITE(6,888) ITIME(4),ITIME(5),ITIME(6)
           WRITE(6,889) ITIME(7),ITIME(8),ITIME(9),ITIME(10)
           STOP
           END
C
```

Extracted from Linear Goal Programming by Marc J. Schniederjans with permission.

201

```
C           SUBROUTINE START READS INPUT AND INITIATES
C            WORKING MATRICES.
          SUBROUTINE START
          IMPLICIT REAL*8(A-H,P-Z)
          INTEGER*4 POS,NEG,END
          INTEGER*4 KSIGN
          INTEGER*2 E,G,L,B
          INTEGER*2 ISIGN
          COMMON NROW,NCOL,NVAR,NPRT,KTEST,ITER
          COMMON /R1/ BASIS(150,300)
          COMMON /R2/ VALC(11,300),VALB(11,150)
          COMMON /R3/ PRHS(150),RHS(150)
          COMMON /I1/ IBASIC(150),JCOL(300)
          DIMENSION ISIGN(300)
          DATA POS,NEG,END/'POS ','NEG ','END '/
          DATA E,G,L,B/'E ','G ','L ','B '/
          READ(5,*) NROW
          READ(5,*) NVAR
          READ(5,*) NPRT
          IF(NROW.LE.O) GO TO 91
          IF(NVAR.LE.O) GO TO 91
          IF(NPRT.LE.O) GO TO 91
          NCOL = NROW + NVAR
          DO 2 I=1,NROW
            DO 1 J=1,NCOL
            BASIS(I,J) = 0.0
            INDEX = J - NVAR
            IF(INDEX.EQ.I) BASIS(I,J) = 1.0
      1     CONTINUE
          IND = I + NCOL
          IBASIC(I) = IND
      2   CONTINUE
          DO 3 J=1,NCOL
          JCOL(J) = J
      3   CONTINUE
          KEND = NPRT + 1
          DO 6 K=1,KEND
            DO 4 J=1,NCOL
            VALC(K,J) = 0.0
      4     CONTINUE
            DO 5 I=1,NROW
            VALB(K,I) = 0.0
      5     CONTINUE
      6   CONTINUE
          KTEST=0
          READ(5,*) (ISIGN(I),I=1,NROW)
          DO 10 I=1,NROW
          IF (ISIGN(I).EQ.E) GO TO 7
```

```
      IF(ISIGN(I).EQ.G) GO TO 8
      IF(ISIGN(I).EG.L) GO TO 9
      IF(ISIGN(I).EQ.B) GO TO 10
      GO TO 92
   7  KTEST =1
      INDEX = I + NVAR
      VALB(1,I) = 1.0
      VALC(1,INDEX) = 1.0
      JCOL(INDEX) = 0
      GO TO 10
   8  INDEX = I + NVAR
      KTEST = 1
      VALC(1,INDEX) = 1.0
      JCOL(INDEX) = 0
      GO TO 10
   9  KTEST = 1
      VALB(1,I) = 1.0
  10  CONTINUE
      IF(KTEST.EQ.1) NPRT = NPRT + 1
  11  READ(5,*) KSIGN,I,K,WGT
      IF(KSIGN.EG.END) GO TO 13
      IF(KTEST.EQ.1) K = K + 1
      IF(KSIGN.EQ.POS) GO TO 12
      IF(KSIGN.NE.NEG) GO TO 94
      INDEX = I + NVAR
      VALC(K,INDEX) = WGT
      GO TO 11
  12  CONTINUE
      VALB(K,I) = WGT
      GO TO 11
  13  CONTINUE
  15  READ(5,*) I,J,AIJ
      IF(I.EQ.O) GO TO 16
      BASIS(I,J) = AIJ
      GO TO 15
  16  CONTINUE
      READ(5,*) (PRHS(I),I=1,NROW)
      DO 23 I=1,NROW
      IF(PRHS(I)) 20,21,22
  20  GO TO 95
  21  PRHS(I) = 1.0E-12
  22  RHS(I) = -PRHS(I)
  23  CONTINUE
      DO 31 J=1,NCOL
      IF(JCOL(J).NE.0) GO TO 31
        DO 30 I=1,NROW
        BASIS(I,J) = 0.0
  30    CONTINUE
```

```
      31 CONTINUE
         RETURN
      91 WRITE(6,1091)
         STOP
      92 WRITE(6,1092)
         STOP
      94 WRITE(6,1094)
         STOP
      95 WRITE(6,1095)
         STOP
    1091 FORMAT (' NUMBER OF CONSTRAINTS, VARIABLES, OR
           PRIORITY LEVEL',/, ' IMPROPERLY ENTERED.')
    1092 FORMAT (' SIGN SYMBOL SOMETHING OTHER THAN E, G,
           L, OR B.')
    1094 FORMAT (' DEVIATION TO BE MINIMIZED NOT POS OR
           NEG')
    1095 FORMAT (' THIS PROGRAM REQUIRES NON-NEGATIVE
           RIGHT HAND SIDES.',/, 'MULTIPLY CONSTRAINT BY
           MINUS ONE.')
         END
C
C        THIS SUBROUTINE PERFORMS THE SIMPLEX OPERATION
C
         SUBROUTINE SIMPLX
         IMPLICIT REAL*8(A-H,P-Z)
         COMMON NROW,NCOL,NVAR,NPRT,KTEST,ITER
         COMMON /R1/ BASIS(150,300)
         COMMON /R2/ VALC(11,300),VALB(11,150)
         COMMON /R3/ PRHS(150),RHS(150)
         COMMON /I1/ IBASIC(150),JCOL(300)
         DIMENSION JFAIL(150),JPICK(300),ZVAL(11,300)
         KEND = NPRT + 1
         DO 16 J=1,NCOL
         JPICK(J) = KEND
      16 CONTINUE
         D0 18 J=1,NCOL
           D0 17 K=1,NPRT
           IF(VALC(K,J).LE.1.0E-10) GO TO 17
           JPICK(J) = K
      17   CONTINUE
      18 CONTINUE
         ITER = 0
       1 KEYROW = 0
         KEYCOL = 0
         KUNACH = 0
           DO 2 I=1,NROW
         JFAIL(I) = 1
       2 CONTINUE
```

```
C
C        IDENTIFY HIGHEST UNACHIEVED PRIORITY
C
      DO 4 K=1,NPRT
        DO 3 I=1,NROW
        IF(VALB(K,I).LE.1.0E-10) GO TO 3
        KUNACH = K
        GO TO 11
 3      CONTINUE
 4    CONTINUE
C
C        IDENTIFY THE MOST NEGATIVE RHS
C
 11   CONTINUE
      RMIN = -1.0E-10
      DO 12 I=1,NROW
      IF(RHS(I).GE.RMIN) GO TO 12
      IF(JFAIL(I).EQ.0) GO TO 12
      KEYROW = I
      RMIN = RHS(I)
 12   CONTINUE
C
C        IF KEYROW EQUALS 0, ALL RHS GREATER THAN OR
C      EQUAL TO 0
      IF(KEYROW.EQ.0) GO TO 30
C
C        PATH FOR NEGATIVE RIGHT HAND SIDE
C
      AIJ = 1.0E-8
      DO 25 M=1,KEND
      L = KEND - M + 1
        DO 24 J=1,NCOL
        IF(JCOL(J).EQ.0) GO TO 24
        IF(JPICK(J).LT.L) GO TO 24
        IF(BASIS(KEYROW,J).LE.AIJ) GO TO 24
        AIJ = BASIS(KEYROW,J)
        KEYCOL = J
 24     CONTINUE
      IF(KEYCOL.GT.0) GO TO 40
 25   CONTINUE
      JFAIL(KEYROW) = 0
      GO TO 11
C
C        PATH FOR NONNEGATIVE RIGHT HAND SIDE
C
 30   CONTINUE
      IF(KUNACH.EQ.0) GO TO 96
      KFIN = KUNACH
```

```
C
C           THE ZJ MATRIX IS DEVELOPED. SINCE BASIS IS
C           NEGATIVE OF CONVENTIONAL, ZJ CALCULATED WILL
C           BE NEGATIVE FOR FAVORABLE VARIABLES.
      DO 33 K=KUNACH,NPRT
        DO 32 J=1,NCOL
        ZVAL(K,J) = 0.0
        IF(JCOL(J).EQ.0) GO TO 32
        IF(JPICK(J).LT.KFIN) GO TO 32
          DO 31 I=1,NROW
          IF(VALB(K,I).LE.1.0E-10) GO TO 31
          IF(DABS(BASIS(I,J)).LE.1.0E-10) GO TO 31
          ZVAL(K,J) = ZVAL(K,J) + VALB(K,I)*BASIS(I,J)
 31       CONTINUE
        ZVAL(K,J) = ZVAL(K,J) + VALC(K,J)
 32     CONTINUE
 33   CONTINUE
      ZVALUE = -1.0E-8
      DO 36 K=KUNACH,NPRT
        DO 35 J=1,NCOL
        IF(JCOL(J).EQ0) GO TO 35
        IF(JPICK(J).LT.KFIN) GO TO 35
        IF(ZVAL(K,J).GE.ZVALUE) GO TO 35
        IF(K.LE.KUNACH) GO TO 39
          M = K - 1
          DO 34 L=1,M
          IF(ZVAL(L,J).GE.1.0E-8) GO TO 35
 34       CONTINUE
 39     CONTINUE
      ZVALUE = ZVAL(K,J)
        KEYCOL = J
 35     CONTINUE
      IF(KEYCOL.GT.0) GO TO 37
      KFIN = KFIN + 1
 36   CONTINUE
      IF(KEYCOL.EQ.0) GO TO 97
 37   THETA = 1.0E9
      DO 38 I=1,NROW
      IF(BASIS(I,KEYCOL).GE.-1.0E-10) GO TO 38
      IF(RHS(I).LE.-1.0E-10) GO TO 38
      IF(RHS(I).LE.1.0E-10) RHS(I) = 1.0E-10
      ZETA = -RHS(I)/BASIS(I,KEYCOL)
      IF(ZETA.GE.THETA) GO TO 38
      THETA = ZETA
      KEYROW = I
 38   CONTINUE
      IF(KEYROW.GT.0) GO TO 40
      GO TO 97
```

```
C
C        SIMPLEX ROUTINE
C
   40  CONTINUE
       PIV = BASIS(KEYROW,KEYCOL)
       DO 43 I=1,NROW
       IF(I.EQ.KEYROW) GO TO 43
       IF(DABS(BASIS(I,KEYCOL)).LE.1.0E-10) GO TO 43
       IF(DABS(RHS(KEYROW)).LE.1.0E-10) GO TO 41
       RHS(I) = RHS(I) - (RHS(KEYROW)/PIV)*BASIS(I,KEYCOL)
   41    DO 42 J=1,NCOL
         IF(J.EQ.KEYCOL) GO TO 42
         IF(DABS(BASIS(KEYROW,J)).LE.1.0E-10) GO TO 42
       BASIS(I,J) = BASIS(I,J) - (BASIS(I,KEYCOL)/
          PIV)*BASIS(KEYROW,J)
   42    CONTINUE
       BASIS(I,KEYCOL) = BASIS(I,KEYCOL)PIV
   43  CONTINUE
       IF(DABS(RHS(KEYROW)).LE.1.0E-10) GO TO 44
       RHS(KEYROW) = -RHS(KEYROW)/PIV
   44  CONTINUE
       DO 45 J=1,NCOL
       IF(J.EQ.KEYCOL) GO TO 45
       IF(DABS(BASIS(KEYROW,J)).LE.1.0E-10) GO TO 45
       BASIS(KEYROW,J) = -BASIS(KEYROW,J)/PIV
   45  CONTINUE
       BASIS(KEYROW,KEYCOL) = 1/PIV
       INDEX = JCOL(KEYCOL)
       JCOL(KEYCOL) = IBASIC(KEYROW)
       IBASIC(KEYROW) = INDEX
       DO 46 K=1,NPRT
       DUMMY = VALB(K,KEYROW)
       IF(DUMMY.GE.1.0E-8) JPICK(KEYCOL) = K
       VALB(K,KEYROW) = VALC(K,KEYCOL)
       VALC(K,KEYCOL) = DUMMY
   46  CONTINUE
       IF(KTEST.NE.1) GO TO 51
       IF(VALC(1,KEYCOL).EQ.0.0) GO TO 51
       JCOL(KEYCOL) = 0
       DO 50 I=1,NROW
       BASIS(I,KEYCOL) = 0.0
   50  CONTINUE
   51  CONTINUE
       ITER = ITER + 1
       GO TO 1
   96  WRITE(6,1096)
   97  RETURN
   98  WRITE(6,1098)
```

```
      STOP
 1096 FORMAT(' ALL GOALS ACHIEVED')
 1098 FORMAT(' THE MODEL IS INFEASIBLE')
      END
C
C     THIS SUBROUTINE REPORTS THE FINAL SOLUTION.
C
      SUBROUTINE FINISH
      IMPLICIT REAL*8(A-H,P-Z)
      COMMON NROW,NCOL,NVAR,NPRT,KTEST,ITER
      COMMON /R1/ BASIS(150,300)
      COMMON /R2/ VALC(11,300),VALB(11,150)
      COMMON /R3/ PRHS(150),RHS(150)
      COMMON /I1/ IBASIC(150),JCOL(300)
      DIMENSION X(150),POSD(150),RNEGD(150)
C
C     THIS SECTION IDENTIFIES AND REPORTS THE VALUES OF
C        ALL MODEL VARIABLES. REAL VARIABLES ARE
C        REPORTED FIRST, THEN DEVIATIONAL VARIABLES
C
      DO 1 J=1,NVAR
      X(J) = 0.0
    1 CONTINUE
      DO 2 I=1,NROW
      POSD(I) = 0.0
      RNEGD(I) = 0.0
    2 CONTINUE
      DO 12 I=1,NROW
      IVAR = IBASIC(I)
      IF(IVAR.GT.NCOL) GO TO 11
      IF(IVAR.GT.NVAR) GO TO 10
      X(IVAR) = RHS(I)
      GO TO 12
   10 CONTINUE
      IND = IVAR - NVAR
      RNEGD(IND) = RHS(I)
      GO TO 12
   11 CONTINUE
      IND = IVAR - NCOL
      POSD(IND) = RHS(I)
   12 CONTINUE
      WRITE(6,1000) ITER
      WRITE(6,1001)
      WRITE(6,1002)
      DO 15 J=1,NVAR
      WRITE(6,1003) J,X(J)
   15 CONTINUE
      WRITE(6,1004)
```

```
      WRITE(6,1005)
      DO 16 I=1,NROW
      WRITE(6,1006) I,PRHS(I),POSD(I),RNEGD(I)
   16 CONTINUE
C
C     THIS SECTION PROVIDES A REPORT OF PRIORITY LEVEL
C        ACHIEVEMENT.
      WRITE(6,1013)
      KTOTAL = NPRT + 1
      DO 52 K=1,NPRT
      KVAL = KTOTAL - K
      M = KVAL
      IF(KTEST.EQ.1) M = KVAL - 1
      ZVALUE = 0.0
        D0 50 I=1,NROW
        IF(VALB(KVAL,I).LE.1.0E-10) GO TO 50
        IF(DABS(RHS(I)).LE.1.0E-10) GO TO 50
        ZVALUE = ZVALUE + VALB(KVAL,I)*RHS(I)
   50 CONTINUE
      IF(KTEST.EQ.0) GO TO 51
      IF(M.GT.0) GO TO 51
      WRITE(6,1015) ZVALUE
      GO TO 52
   51 WRITE(6,1014) M,ZVALUE
   52 CONTINUE
      RETURN
 1000 FORMAT (I6,'      ITERATIONS')
 1001 FORMAT (' DECISION VARIABLES')
 1002 FORMAT (/,' VARIABLE              VALUE')
 1003 FORMAT (3X,I5,3X,F15.5)
 1004 FORMAT (///,' ANALYSIS OF DEVIATIONS FROM GOALS')
 1005 FORMAT (/,' ROW',8X,'RHS-VALUE',10X,'POSITIVE
        DEVIATION',6X, 'NEGATIVE DEVIATION')
 1006 FORMAT (I4,3F20.5)
 1013 FORMAT (///,' ANALYSIS OF THE OBJECTIVE FUNCTION',//,'
        PRIORITY',9 X,'UNDERACHIEVEMENT')
 1014 FORMAT (I3,9X,F20.5)
 1015 FORMAT (' ARTIFICIAL',F20.5)
      END
```

Appendix B

LGP Micro Computer Program

```
 10    REM  SET UP PROBLEM AND FLAGS
 20    HOME : CLEAR
 30  D$ = CHR$ (4)
 40    INPUT "DO YOU WANT INSTRUCTIONS? ";RP$
 50    IF RP$ = "Y" THEN GOSUB 2150
 60    IF RP$ = "Y" OR RP$ = "N" THEN GOTO 80
 70    PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 40
 80    PRINT
 90    INPUT "IS YOUR PROBLEM ALREADY ON FILE? ";RR$
100    IF RR$ = "Y" OR RR$ = "N" THEN GOTO 120
110    PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 90
120    PRINT
130    INPUT "NAME YOUR PROBLEM. ";PR$
140    IF RR$ = "Y" THEN GOSUB 470
150    IF RR$ = "N" THEN GOSUB 790
160    IF RR$ = "N" THEN GOSUB 5740
170    IF RR$ = "N" THEN GOTO 230
180    IF RR$ = "Y" THEN INPUT "DO YOU WANT TO CHANGE IT?
         ";PF$
190    IF PF$ = "Y" THEN GOSUB 6020
200    IF PF$ = "Y" THEN GOTO 230
210    IF PF$ = "N" THEN GOTO 230
220    PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 180
230    PRINT
240    INPUT "DO YOU WANT PRINTOUT? ";PO$
250    IF PO$ = "Y" THEN PRINT D$"PR#1": PRINT PR$: PRINT :
         PRINT D$"PR#0"
```

Extracted from <u>Linear Goal Programming</u>
by Marc J. Schniederjans with permission 211

```
260    IF PO$ = "Y" THEN INPUT "INCLUDING TABLEAU? ";TB$: IF
          TB$ = "Y" THEN GOTO 300
270    IF PO$ = "Y" THEN GOTO 330
280    IF PO$ = "N" THEN GOTO 330
290    PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 240
300    INPUT "(A)LL OR JUST (F)IRST? ";QQ$
310    IF QQ$ = "A" OR QQ$ = "F" THEN GOTO 330
320    PRINT "A OR F ONLY. TRY AGAIN.": GOTO 300
330    PRINT
340    GOSUB 2070
350    GOSUB 5480
360    IF QQ$ = "F" AND TC > 0 THEN GOTO 420
370    IF QQ$ = "F" THEN GOSUB 5620
380    IF QQ$ = "F" THEN GOSUB 5020
390    IF QQ$ = "F" THEN GOTO 420
400    IF QQ$ = "A" AND TC = 0 THEN GOSUB 5620
410    IF QQ$ = "A" THEN GOSUB 5020
420    GOSUB 3210
430    GOSUB 3710
440    GOSUB 4350
450    GOSUB 4690
460    GOTO 350
470    REM   READ FILE FROM DISK
480    PRINT D$"OPEN";PR$;",L300"
490    PRINT D$"READ";PR$;",R";0
500    INPUT NU: INPUT MC; INPUT P: INPUT N$
510    DIM A(MC,NU + (2 * MC) + 1),CZ(P,NU + (2 * MC) +
          1),C(NU + (2 * MC)),B(MC), WC(NU + (2 * MC));WB(MC)
520    DIM N$(NU)
530    DIM Y$(MC),DI(NU + 2 * MC)
540    DIM DB(NU + 2 * MC)
550    FOR I = 1 TO MC
560    PRINT D$"READ";PR$;",R";I
570    FOR J = 1 TO NU + (2 * MC) + 1
580    INPUT A(I,J)
590    NEXT J
600    NEXT I
610  I = MC + 1
620    PRINT D$"READ";PR$;",R";I
630    FOR J = 1 TO NU + (2* MC)
640    INPUT C(J)
650    NEXT J
660  I = I + 1
670    PRINT D$"READ";PR$",R";I
680    FOR J = 1 TO NU + (2 * MC)
690    INPUT WC(J)
700    NEXT J
710    IF N$ < > "Y" THEN GOTO 770
```

```
720  I = I + 1
730    PRINT D$"READ";PR$;",R";I
740    FOR J = 1 TO NU
750    INPUT N$(J)
760    NEXT J
770    PRINT D$"CLOSE";PR$;" "
780    RETURN
790    REM   DATA ENTRY ROUTINE
800    HOME : PRINT "READY TO ENTER DATA.": PRINT
810    PRINT : INPUT "NUMBER OF UNKNOWNS ";NU
820    PRINT
830    INPUT "WANT TO NAME VARIABLES? ";N$
840    PRINT
850    IF N$ = "N" THEN GOTO 920
860    IF N$ = "Y" THEN GOTO 880
870    PRINT "TRY AGAIN.": GOTO 830
880    DIM N$(NU): FOR I = 1 TO NU
890    PRINT "VARIABLE X";I;: INPUT " REPRESENTS ";N$(I)
900    NEXT I
910    PRINT
920    PRINT "REMEMBER ONLY EQUATIONS WITH"
930    PRINT "DECISION VARIABLES COUNT IN"
940    PRINT "ANSWERING NEXT QUESTION."
950    INPUT "NUMBER OF CONSTRAINTS ";MC
960    PRINT
970    INPUT "NO. OF DEVIATIONAL VARIABLES? ";DV
980    PRINT
990    INPUT "NUMBER OF PRIORITIES ";P
1000   PRINT
1010   DIM A(DV,NU + (2 * DV) + 1)
1020   DIM CZ(P,NU + 2 * DV +1)
1030   DIM C(NU + 2 * DV)
1040   DIM B(DV)
1050   DIM WC(NU + (2 * DV))
1060   DIM WB(DV)
1070   DIM Y$(DV),DI(NU + 2 * DV)
1080   FOR I = 1 TO MC
1090   HOME : PRINT "IF ONLY 3 OR 4 UNKNOWNS IN PROBLEM"
1100   PRINT "ANSWER NEXT QUESTION WITH 'A'."
1110   PRINT "IT WILL BE FASTER TO ENTER ALL."
1120   PRINT : PRINT "TYPE IN NUMBER OF VARIABLES"
1130   PRINT "WHICH APPEAR IN EQUATION ";I
1140   INPUT "OR A FOR (A)LL. ";AA$
1150   IF AA$ = "A" THEN GOTO 1240
1160 AA = VAL (AA$)
1170   FOR J = 1 TO AA
1180   PRINT "ENTER SUBSCRIPT OF UNKNOWN ";J
1190   PRINT "IN EQUATION ";I;" ";: INPUT BB$
```

```
1200   PRINT "ENTER VALUE OF X";BB$;" ";: INPUT A$
1210  A(I, VAL (BB$)) = VAL (A$)
1220   NEXT J
1230   GOTO 1280
1240   FOR J = 1 TO NU
1250   PRINT "ENTER VALUE OF X";J;" ";: INPUT A$
1260  A(I,J) = VAL (A$)
1270   NEXT J
1280   PRINT
1290   PRINT "IS POS. DEV. VAR. ALLOWED IN"
1300   PRINT "EQUATION ";I; " ";: INPUT CC$
1310   IF CC$ = "N" THEN GOTO 1350
1320   IF CC$ = "Y" THEN GOTO 1340
1330   PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 1280
1340  A(I,NU + I) = 1:A(I,NU + DV + I) = - 1: GOTO 1360
1350  A(I,NU + I) = 1
1360   PRINT "RHS FOR EQUATION ";I;" ";: INPUT A$
1370  A(I,NU + 2 * DV + 1) = VAL (A$)
1380   IF A(I,NU + 2 * DV + 1) > = 0 THEN GOTO 1420
1390   FOR J = 1 TO NU + 2 * DV + 1
1400  A(I,J) = A(I,J) * - 1
1410   NEXT J
1420   PRINT
1430   PRINT "EQUATION ";I;" READS:"
1440   PRINT
1450   FOR J = 1 TO NU
1460   PRINT A(I,J);"X";J;" + ";
1470   NEXT J
1480   PRINT "D";I;"—";
1490   IF CC$ = "Y" THEN PRINT " — D";I;"+";
1500   PRINT " = ";
1510   PRINT A(I,NU + 2 * MC + 1)
1520   PRINT "IS IT RIGHT?";: INPUT A$
1530   IF A$ = "Y" THEN GOTO 1560
1540   IF A$ = "N" THEN HOME: PRINT "REENTER EQUATION ";I;".":
         GOTO 1170
1550   PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 1520
1560   NEXT I
1570   IF DV > MC THEN PRINT "YOUR DEV. VARS. NOT IN OTHER
         CONSTRAINTS ARE:": GO TO 1590
1580   GOTO 1840
1590   PRINT "D";MC + 1;" TO D";DV
1600   FOR I = MC + 1 TO DV
1610   PRINT
1620   PRINT "HOW MANY OTHER DEV. VARS. APPEAR"
1630   PRINT "IN EQUATION FOR D";I;: INPUT SS
1640   IF SS = 0 THEN NEXT I
1650   FOR J = 1 TO SS
```

```
1660   PRINT "EQUATION NUMBER OF DEV. VAR. ";J;" IN D";I;:
         INPUT TT
1670   INPUT "(P)OS OR (N)EG DEV. VAR.? ";VV$
1680   IF VV$ = "P" THEN GOTO 1690
1681   IF VV$ = "N" THEN GOTO 1690
1682   GOTO 1670
1690   INPUT "(P)OS OR (N)EG VALUE? ";WW$
1700   IF WW$ = "P" THEN GOTO 1710
1701   IF WW$ = "N" THEN GOTO 1710
1702   GOTO 1690
1710   IF VV$ = "P" AND WW$ = "P" THEN A(I,NU + DV + TT) = 1
1720   IF VV$ = "P" AND WW$ = "N" THEN A(I,NU + DV + TT) =
         - 1
1730   IF VV$ = "N" AND WW$ = "P" THEN A(I,NU + TT) = 1
1740   IF VV$ = "N" AND WW$ = "N" THEN A(I,NU + TT) = - 1
1750 A(I,NU + I) = 1:A(I,NU + DV + I) = - 1
1760   NEXT J
1770   PRINT "RHS FOR THIS EQUATION? ";: INPUT A(I,NU + 2 *
         DV + 1)
1780   IF A(I,NU + 2 * DV + 1) > = 0 THEN GOTO 1820
1790   FOR J = 1 TO NU + 2 * MC + 1
1800 A(I,J) = A(I,J) * - 1
1810   NEXT J
1820   NEXT I
1830 MC = DV
1840 L = 1
1850   PRINT
1860   FOR J = NU + 1 TO NU + MC
1870   PRINT "PRIORITY ASSOCIATED WITH D";L;"-";: INPUT A$
1880   IF A$ = CHR$ (13) THEN C(J) = 0: GOTO 1900
1890 C(J) = VAL (A$)
1900   INPUT "WEIGHT FOR THE DEV.VAR.";A$
1910   IF A$ = CHR$ (13) THEN WC(J) = 0: GOTO 1930
1920 WC(J) = VAL (A$)
1930 L = L + 1
1940   NEXT J
1950 L = 1
1960   FOR J = NU + MC + 1 TO NU + (2 * MC)
1970   PRINT "PRIORITY ASSOCIATED WITH D";L;"+";: INPUT A$
1980   IF A$ = CHR$ (13) THEN C(J) = 0: GOTO 2000
1990 C(J) = VAL (A$)
2000   INPUT "WEIGHT FOR THE DEV.VAR.";A$
2010   IF A$ = CHR$ (13) THEN WC(J) = 0: GOTO 2030
2020 WC(J) = VAL (A$)
2030 L = L + 1
2040   PRINT
2050   NEXT J
2060   RETURN
```

```
2070   REM  CB IN INITIAL TABLEAU
2080 I = 1
2090   FOR J = NU + 1 TO NU + MC
2100 B(I) = C(J)
2110 WB(I) = WC(J):Y$(I) = "D" + STR$ (J - NU) + "-"
2120 I = I + 1
2130   NEXT J
2140   RETURN
2150   REM  INSTRUCTIONS
2160   HOME
2170   PRINT "THIS PROBLEM SOLVES A GOAL PROGRAMMING"
2180   PRINT "PROBLEM BUT NEEDS A LITTLE INTRO."
2190   PRINT
2200   PRINT "TO THIS END, 'SCHNIEDERJANS'"
2210   PRINT "EXAMPLE' HAS BEEN PROVIDED"
2220   PRINT "ON DISKETTE. IT IS PRETTY STRAIGHT— "
2230   PRINT "FORWARD EXCEPT FOR THE"
2240   PRINT "PRIORITIES WHERE, FOR MODELING"
2250   PRINT "REASONS, THE ARTIFICIAL, OR 0,
2260   PRINT "PRIORITY BECOMES PRIORITY 1 AND"
2270   PRINT "ALL THE OTHER, STATED, PRIORITIES"
2280   PRINT "SLIP DOWN ONE. INSTEAD OF 'FOUR'"
2290   PRINT "THE ANSWER TO THE QUESTION, 'NUMBER"
2230   PRINT "OF PRIORITIES?' IS 'FIVE'.
2310   PRINT "THE OPERATOR MUST MAKE THIS"
2320   PRINT "CONVERSION."
2330   PRINT: PRINT "PRESS ANY KEY TO CONTINUE.";: GET Q$
2340   HOME
2350   PRINT "ANOTHER PROGRAM QUIRK OCCURS IF THERE"
2360   PRINT "ARE NO UNKNOWNS (JUST DEV. VARS.)
2365   PRINT "IN AN EQUATION."
2370   PRINT "WHEN IT ASKS FOR NUMBER OF CONSTRAINTS"
2380   PRINT "ONLY ENTER THE NUMBER IN WHICH"
2390   PRINT "UNKNOWNS APPEAR."
2400   PRINT "DON'T TRY TO SUBSTITUTE DEVIATIONAL"
2410   PRINT "VARIABLES INTO CONSTRAINTS LIKE"
2420   PRINT "YOU MIGHT TRY TO DO TO LIMIT OVERTIME."
2430   PRINT "INSTEAD, PROGRAM WILL CREATE"
2440   PRINT "SEPARATE EQUATIONS FOR STAND ALONE"
2450   PRINT "DEVIATIONAL VARIABLES AS YOU"
2460   PRINT "ANSWER FURTHER QUESTIONS ABOUT"
2470   PRINT "DEV. VARS. IN EXCESS OF CONSTRAINTS."
2480   PRINT: PRINT "PRESS ANY KEY TO CONTINUE.";: GET Q$
2490   HOME
2660   PRINT "TABLEAUX ARE NOT NEAT. THEY"
2670   PRINT "ARE JAMMED TOGETHER IN THE ATTEMPT"
2680   PRINT "TO GET ALL ON THE FEWEST PRINTER"
2690   PRINT "LINES. THE OPTION TO PRINT"
```

```
2700    PRINT "THEM IS PROVIDED FOR CHECKOUT"
2710    PRINT "PURPOSES ONLY. READ STARTING WITH"
2720    PRINT "RHS COLUMN ON LEFT IN A(I,J) PORTION"
2730    PRINT "AND AMOUNT OF REMAINING PRIORITY"
2740    PRINT "TO FILL ON LEFT IN ZJ-CZ PORTION."
2750    PRINT
2760    PRINT "CB AND CJ ARE NOT PRINTED."
2770    PRINT "YOU CAN DETERMINE WHAT THEY ARE AT"
2780    PRINT "END OF RUN BY ASKING FOR PRINT"
2790    PRINT "OF 'C(I)' AND 'WC(I)' FOR CJ"
2800    PRINT "WHERE 'I' IS COLUMN NUMBER, 'C(I)'"
2810    PRINT "IS THE SUBSCRIPT PRIORITY AND 'WC(I)'"
2820    PRINT "IS THE WEIGHT ASSIGNED."
2830    PRINT "THE SAME IS TRUE FOR 'CB' USING"
2840    PRINT "'B(I)' AND 'WB(I)'
2850    PRINT "WHERE 'I' IS THE ROW NUMBER."
2860    PRINT "PRESS ANY KEY TO CONTINUE. ';: GET Q$
2870    HOME
2880    PRINT "THE OTHER IMPORTANT VARIABLE NAMES"
2890    PRINT "ARE 'A(I,J)' FOR A(I,J)"
2900    PRINT "AND 'CZ(I,J)' FOR ZJ-CJ"
2910    PRINT "WHERE 'I' IS 1 TO NUMBER OF DEV."
2920    PRINT "VARS. IN A(I,J) AND 1 TO NUMBER"
2930    PRINT "OF PRIORITIES IN CZ(I,J) AND"
2940    PRINT "J IS 1 TO NUMBER OF CONSTRAINTS PLUS"
2950    PRINT "TWICE THE NUMBER OF DEV. VARS. PLUS"
2960    PRINT "ONE(TO INCLUDE RHS) IN BOTH CASES."
2970    PRINT
2980    PRINT "PRESS ANY KEY TO CONTINUE.";: GET Q$
2990    HOME
3000    PRINT "TO RUN PROGRAM, THE FIRST QUESTION"
3010    PRINT "GOT YOU HERE. THE NEXT WILL"
3020    PRINT "ASK WHETHER OR NOT YOUR PROBLEM"
3030    PRINT "IS ALREADY ON FILE (THE DISK)."
3040    PRINT "ANSWER 'Y' OR 'N' AS APPROPRIATE."
3050    PRINT "THE NEXT QUESTION ASKS YOU TO"
3060    PRINT "NAME YOUR PROBLEM."
3070    PRINT "BE CAREFUL NOT TO USE A NAME"
3080    PRINT "OF A FILE ALREADY ON DISK."
3090    PRINT: FLASH : PRINT "IT WILL GET WIPED OUT.": NORMAL
3100    PRINT: PRINT "TO CHANGE A PROBLEM ON DISK ANSWER
          YES"
3110    PRINT "TO NEXT QUESTION.  TO RERUN A PROBLEM"
3120    PRINT "ALREADY ON DISK ANSWER NO TO"
3130    PRINT "THIS QUESTION AND PROGRAM WILL"
3140    PRINT "MERELY REGURGITATE A PREVIOUSLY"
3150    PRINT "STORED PROBLEM AFTER YOU ANSWER"
3160    PRINT "THE FINAL QUESTION ON PRINTING."
```

45


```
3170    PRINT : PRINT "I THINK THAT SHOULD BE ENOUGH"
3180    PRINT "TO ALLOW YOU TO RUN PROGRAM."
3190    PRINT : PRINT "PRESS ANY KEY TO START";: GET Q$
3200    HOME : RETURN
3210    REM    CHECK FOR DONE
3220  Z = 0
3230    FOR K = 1 TO P
3240    IF CZ(K,NU + 2 * MC + 1) > 0 THEN GOTO 3270
3250    NEXT K
3260    GOTO 5230
3270    IF K > 1 AND Z = 0 THEN GOTO 3550
3280    IF K > 1 THEN GOTO 3380
3290    FOR J = 1 TO NU + 2 * MC
3300    IF CZ(K,J) > 0 AND J < NU + 1 THEN RETURN
3310    IF CZ(K,J) > 0 AND K < C(J) THEN RETURN
3320    NEXT J
3330    NEXT K
3340    PRINT "NO POS VALUES IN PRIORITY 1."
3350    PRINT "PROBLEM IS INFEASIBLE."
3360    PRINT "PRESENT STATUS IS:
3370    PRINT : GOTO 5230
3380    FOR J = 1 TO NU + 2 * MC
3390    IF CZ(K,J) > 0 THEN GOTO 3430
3400    NEXT J
3410    NEXT K
3420    GOTO 5230
3430    FOR M = 1 TO K - 1
3440    IF CZ(M,J) < 0 THEN GOTO 3470
3450    NEXT M
3460    GOTO 3500
3470    NEXT J
3480    NEXT K
3490    GOTO 5230
3500    IF J < NU + 1 THEN RETURN
3510    IF K < C(J) THEN RETURN
3520    NEXT J
3530    NEXT K
3540    GOTO 5230
3550  Z = Z + 1
3560    FOR I = 1 TO K - 1
3570    IF CZ(I,NU + 2 * MC + 1) = 0 THEN GOTO 3590
3580    NEXT I
3590    FOR J = 1 TO NU + 2 * MC
3600    IF CZ(I,J) > 0 THEN GOTO 3640
3610    NEXT J
3620    NEXT I
3630    GOTO 3380
3640    IF I = 1 THEN RETURN
```

```
3650    FOR L = 1 TO M - 1
3660    IF CZ(L,J) < 0 THEN GOTO 3610
3670    NEXT L
3680    IF J < NU + 1 THEN RETURN
3690    IF I < C(J) THEN RETURN
3700    GOTO 3610
3710    REM   DETERMINE PIVOT COLUMN
3720  Z = 0:I = 0
3730    FOR K = 1 TO P
3740    FOR J = 1 TO NU + 2 * MC
3750    IF CZ(K,J) < = 0 THEN GOTO 3820
3760    IF CZ(K,J) < Z THEN GOTO 3820
3770    IF CZ(K,J) = > Z THEN GOSUB 3830
3780    IF CV = 1 THEN CV = 0: GOTO 3820
3790    IF CZ(K,J) = 0 THEN GOTO 3810
3800    IF CZ(K,J) > Z THEN Z = CZ(K,J):I = 1:DI(1) = J: GOTO
          3820
3810  I = I + 1:DI(I) = J
3820    NEXT J: GOTO 3920
3830  CV = 0
3840    IF K = 1 THEN RETURN
3850    IF K = 2 THEN GOTO 3900
3860    FOR JJ = K - 1 TO 1 STEP - 1
3870    IF CZ(JJ,J) < 0 THEN CV = 1: RETURN
3880    NEXT JJ
3890    RETURN
3900    IF CZ(1,J) < 0 THEN CV = 1
3910    RETURN
3920    IF I = 1 THEN M = DI(1):TB = K: GOTO 4300
3930    IF I > 1 THEN GOTO 3950
3940    NEXT K: GOTO 5230
3950  Z = 0:II = 0
3960    IF K > = P THEN GOSUB 4310
3970  M = DI(1 + QZ):TB = P
3980    FOR L = K + 1 TO P
3990    FOR J = 1 TO I
4000    IF CZ(L,DI(J)) < = 0 THEN GOTO 4070
4010    IF CZ(L,DI(J)) < Z THEN GOTO 4070
4020    IF CZ(L,DI(J)) > = Z THEN GOSUB 4080
4030    IF CV = 1 THEN CV = 0: GOTO 4070
4040    IF CZ(L,DI(J)) = Z THEN GOTO 4060
4050    IF CZ(L,DI(J)) > Z THEN Z = CZ(L,DI(J)):II = 1:DB(II) = DI(J):
          GOTO 4070
4060  II = II + 1:DB(II) = DI(J)
4070    NEXT J: GOTO 4160
4080  CV = 0
4090    IF L = 2 THEN GOTO 4140
4100    FOR JJ = L - 1 TO L STEP - 1
```

```
4110    IF CZ(JJ,DI(J)) < 0 THEN CV = 1: RETURN
4120    NEXT JJ
4130    RETURN
4140    IF CZ(1,DI(J)) < 0 THEN CV = 1
4150    RETURN
4160    IF II > 0 THEN I = II
4170    IF Z = 0 THEN GOTO 4270
4180    IF II = 1 THEN M = DB(1):TB = L: GOTO 4300
4190    IF L > = P AND II = 0 THEN GOSUB 4310
4200  M = DI(QZ + 1):TB = P: RETURN
4210    IF L > = P THEN GOSUB 4330
4220  M = DB(1 + QZ):TB = P: RETURN
4230    FOR J = 1 TO II
4240  DI(J) = DB(J)
4250    NEXT J
4260  Z = 0:II = 0
4270    NEXT L
4280    GOSUB 4310
4290  M = DI(1):TB = 1: RETURN
4300    RETURN
4310    IF I = QZ THEN PRINT "ALL PIVOTS TRIED": PRINT "NO
          WAY OUT OF LOOP": PRINT "CURRENT STATUS IS:":
          GOTO 5230
4320    RETURN
4330    IF II = QZ THEN PRINT "ALL PIVOTS TRIED": PRINT "NO
          WAY OUT OF LOOP": PRINT "CURRENT STATUS IS:":
          GOTO 5230
4340    RETURN
4350    REM   DETERMINE PIVOT ROW
4360  P2 = P1:P1 = PC:PC = M:M = 1
4370    IF QQ$ = "F" AND TC > 0 THEN GOTO 4390
4380    IF TB$ = "Y" THEN PRINT D$"PR#1"
4390    PRINT "PIVOT COLUMN = ";PC
4400    IF TB$ = "Y" THEN PRINT D$"PR#0"
4410  J = 0:M = 0:DR = 0
4420    FOR I = 1 TO MC
4430    IF A(I,PC) < = 0 THEN DI(I) = 0: GOTO 4500
4440  DI(I) = A(I,(NU + 2 * MC + 1)) / A(I,PC)
4450    IF DI(I) < 0 THEN GOTO 4500
4460    IF DR = 0 THEN GOTO 4480
4470    IF DI(I) > DR THEN GOTO 4500
4480    IF DI(I) < DR OR DR = 0 THEN M = I:DR = DI(I):J = 1:DB(J)
          = I: GOTO 4500
4490    IF DI(I) = DR AND DR > 0 THEN J = J + 1:DB(J) = I
4500    NEXT I
4510    IF J = 0 THEN PRINT "THE SOLUTION IS UNBOUNDED.":
          END
4520    IF J = 1 THEN GOTO 4680
```

```
4530 DR = O:K = 0
4540   FOR I = 1 TO J
4550   IF B(DB(I)) < 0 THEN GOTO 4600
4560   IF DR = 0 THEN GOTO 4580
4570   IF B(DB(I)) > DR THEN GOTO 4600
4580   IF B(DB(I)) < DR OR DR = 0 THEN M = DB(I):DR =
         B(DB(I)):K = 1:DI(K) = DB(I): GOTO 4600
4590   IF B(DB(I)) = DR AND DR > 0 THEN K = K + 1:DI(K) =
         DB(I)
4600   NEXT I
4610   IF K = 0 OR K = 1 THEN GOTO 4680
4620 DR = 0:L = 0
4630   FOR I = 1 TO K
4640   IF WB(DI(I)) < DR THEN GOTO 4670
4650   IF WB(DI(I)) > DR THEN M = DI(I):DR = WB(DI(I)):L =
         1:DB(L) = DI(I): GOTO 4670
4660   IF WB(DI(I)) = DR THEN L = L + 1:DB(L) = DI(I)
4670   NEXT I
4680   RETURN
4690   REM  DETERMINE COEFFICIENTS FOR NEXT TABLEAU
4700   IF QQ$ = "F" AND TC > 0 THEN GOTO 4720
4710   IF TB$ = "Y" THEN PRINT D$PR#1"
4720 P4 = P3:P3 = PR:PR = M:M = 1:DI = A(PR,PC)
4730   PRINT "PIVOT ROW = ";PR: PRINT
4740   IF TB$ = "Y" THEN PRINT D$"PR#0"
4750   IF PC = P2 AND PR = P4 THEN PRINT "IN A LOOP, TRYING
         AGAIN.": GOTO 4960
4760 QZ = 0
4770   FOR J = 1 TO NU + (2 * MC) + 1
4780 A(PR,J) = A(PR,J) / DI
4790   NEXT J
4800   FOR I = 1 TO MC: GOTO 4810
4810   IF I = PR THEN NEXT I: GOTO 4870
4820 DI = A(I,PC)
4830   FOR J = 1 TO NU + (2 * MC) + 1
4840 A(I,J) = A(I,J) - (DI * A(PR,J))
4850   NEXT J
4860   NEXT I
4870 B(PR) = C(PC)
4880 WB(PR) = WC(PC)
4890 TC = TC + 1
4900   IF PC < = NU THEN Y$(PR) = "X" + STR$ (PC): RETURN
4910   IF PC > NU AND PC < NU + MC + 1 THEN Y$(PR) = "D" +
         STR$ (PC - NU) + "-" :RETURN
4920 Y$(PR) = "D" + STR$ (PC - NU - MC) + "+"
4930   RETURN
4940   REM  LAST THREE ROWS BROUGHT NEW VARIABLE
         NAMES
```

```
4950   REM  AND WEIGHTS/PRIORITIES INTO BASIS
4960  QZ = QZ + 1
4970   FOR J = 1 TO NU + 2 * MC
4980   IF J = PC THEN GOTO 5000
4990   IF CZ(TB,J) = CZ(TB,PC) THEN GOTO 430
5000   NEXT J
5010   GOTO 5230
5020   REM   TABLEAU PRINTOUT
5030   IF TB$ = "Y" THEN PRINT D$"PR#1"
5040   PRINT
5050   PRINT "COEFFICIENTS IN TABLEAU:"
5060   PRINT
5070   FOR I = 1 TO MC
5080   PRINT Y$(I);" ";A(I,NU + (2 * MC) + 1);" ";
5090   FOR J = 1 TO NU + (2 * MC)
5100   PRINT A(I,J);" ";
5110   NEXT J: PRINT : NEXT I
5120   PRINT
5130   PRINT "VALUES IN ZJ-CJ:"
5140   PRINT
5150   FOR K = P TO 1 STEP - 1
5160   PRINT "P";K;" ";
5170   PRINT CZ(K,NU + (2 * MC) + 1);" ";
5180   FOR J = 1 TO NU + (2 * MC)
5190   PRINT CZ(K,J);" ";
5200   NEXT J: PRINT : NEXT K
5210   IF TB$ = "Y" THEN PRINT D$"PR#0"
5220   RETURN
5230   REM  SOLUTION PRINTOUT
5240   IF PO$ = "Y" THEN PRINT D$"PR#1"
5250   PRINT
5260   PRINT "SOLUTION VARIABLES ARE:"
5270   PRINT
5280   FOR I = 1 TO MC
5290  QQ = LEN (Y$(I))
5300   IF N$ = "Y" AND LEFT$ (Y$(I),1) = "X" THEN PP$ = RIGHT$
         (Y$(I),QQ - 1):PP = VAL (PP$): PRINT N$(PP);: HTAB 20:
         PRINT A(I,NU + (2 * MC) + 1): GOTO 5320
5310   PRINT Y$(I);: HTAB 20: PRINT A(I,NU + (2 * MC) + 1): GOTO
         5320
5320   NEXT I
5330   PRINT
5340  Z = 0
5350   PRINT "UNACHIEVED GOALS ARE:"
5360   PRINT
5370   FOR K = 1 TO P
5380   IF CZ(K,NU + (2 * MC) + 1) = 0 THEN GOTO 5400
5390   PRINT "P";K;: HTAB 20: PRINT CZ(K,NU + (2 * MC) + 1):Z =
         1
```

```
5400    NEXT K
5410    IF Z = 0 THEN PRINT "NONE": PRINT : PRINT: PRINT
5420    IF PO$ = "Y" THEN PRINT D$"PR#0"
5430    PRINT "PRESS 'R' TO DO ANOTHER PROBLEM"
5440    INPUT "OR PRESS 'Q' TO QUIT.";X$
5450    IF X$ = "Q" THEN END
5460    IF X$ = "R" THEN HOME: GOTO 10
5470    PRINT "YOU HIT A WRONG KEY.": GOTO 5430
5480    REM   ZJ CALCULATION
5490  Z = 0
5500    FOR K = 1 TO P
5510    FOR J = 1 TO NU + (2 * MC) + 1
5520    FOR I = 1 TO MC
5530    IF B(I) = K THEN Z = Z + WB(I) * A(I,J)
5540    NEXT I
5550  CZ(K,J) = Z
5560  Z = 0
5570    IF J = NU + 2 * MC + 1 THEN GOTO 5590
5580    IF C(J) = K THEN CZ(K,J) = CZ(K,J) = WC(J)
5590    NEXT J
5600    NEXT K
5610    RETURN
5620    REM   PRIORITY AND WEIGHT PRINTOUT
5630    IF TB$ = "Y" THEN PRINT D$"PR#1"
5640    IF TB$ = "N" THEN RETURN
5650    IF QQ$ = "F" AND TC > 0 THEN RETURN
5660    IF TC = 0 THEN I = 1:K = 1: PRINT : GOTO 5680
5670    RETURN
5680    FOR J = NU + 1 TO NU + MC: PRINT "D";I;"- ";C(J);: HTAB
          30: PRINT "WT ";WC(J)
5690  I = I + 1: NEXT J
5700    FOR J = NU + MC + 1 TO NU + (2 * MC): PRINT "D";K;"+
          ";C(J);: HTAB 30: PRINT "WT ";WC(J)
5710  K = K + 1: NEXT J
5720    IF TB$ = "Y" THEN PRINT D$"PR#0"
5730    RETURN
5740    REM   WRITE SUBROUTINE
5750    PRINT D$"OPEN";PR$;",L300"
5760    FOR I = 1 TO MC
5770    PRINT D$"WRITE";PR$;",R";I
5780    FOR J = 1 TO NU + (MC * 2) + 1
5790    PRINT A(I,J)
5800    NEXT J
5810    NEXT I
5820  I = MC + 1
5830    PRINT D$"WRITE";PR$;",R";I
5840    FOR J = 1 TO NU + (2 * MC)
5850    PRINT C(J)
5860    NEXT J
```

51


```
5870 I = I + 1
5880   PRINT D$"WRITE";PR$;",R";I
5890   FOR J = 1 TO NU + (2 * MC)
5900   PRINT WC(J)
5910   NEXT J
5920 I = I + 1
5930   IF N$ = "Y" THEN PRINT D$"WRITE";PR$;",R";I: GOTO 5950
5940   GOTO 5980
5950   FOR J = 1 TO NU
5960   PRINT N$(J)
5970   NEXT J
5980   PRINT D$"WRITE";PR$;",R";0
5990   PRINT NU: PRINT MC: PRINT P: PRINT N$
6000   PRINT D$"CLOSE";PR$;" "
6010   RETURN
6020   REM   CHANGE SUBROUTINE
6030   HOME : PRINT
6040   PRINT "IF YOU WANT TO CHANGE PRIORITIES"
6050   PRINT "ANSWER TO EQUATION NO. IS ";MC + 1
6060   PRINT "IF YOU WANT TO CHANGE WEIGHTS"
6070   PRINT "ANSWER TO EQUATION NO. IS ";MC + 2
6080   PRINT "IF YOU WANT TO ADD OR DELETE"
6090   PRINT "PRIORITIES OR CONSTRAINTS"
6100   PRINT "RESET AND RUN A NEW PROBLEM."
6110   PRINT
6120   PRINT "ANSWER QUESTIONS ABOUT WHAT"
6130   PRINT "UNKNOWN OR DEV. VAR. WITH THE "
6140   PRINT "FULL NAME (E.G. 'X1' OR 'D1+')."
6150   PRINT
6160   INPUT "WHAT EQUATION NUMBER? ";KK
6170   IF KK > MC + 2 THEN PRINT "NOT THAT MANY
         EQUATIONS. ONLY ";MC: GOTO 6190
6180   GOTO 6200
6190   PRINT "PLUS 2 FOR WEIGHTS AND PRIORITIES.": GOTO
         6160
6200   PRINT
6210   IF KK = MC + 1 THEN GOTO 6530
6220   IF KK = MC + 2 THEN GOTO 6720
6230   IF KK < = MC THEN PRINT "IF YOU WANT TO CHANGE
         RHS,"
6240   PRINT "THE ANSWER TO 'WHAT VARIABLE' IS"
6250   PRINT "'RHS'."
6260   PRINT
6270   INPUT "WHAT VARIABLE? "?JJ$
6280   PRINT "WHAT IS NEW VALUE OF ";JJ$;: INPUT JJ: PRINT
6290   IF JJ$ = "RHS" THEN A(KK,NU + 2 * MC + 1) = JJ: GOTO
         6910
6300   IF LEFT$ (JJ$,1) = "X" THEN GOTO 6350
```

```
6310    IF LEFT$ (JJ$,1) = "D" AND RIGHT$ (JJ$,1) = "+" THEN
          GOTO 6410
6320    IF LEFT$ (JJ$,1) = "D" AND RIGHT$ (JJ$,1) = "-" THEN
          GOTO 6470
6330    PRINT "DIDN'T ANSWER WITH XI, DI+, DI- OR"
6340    PRINT "RHS. TRY AGAIN.": GOTO 6270
6350    IF LEN (JJ$) = 2 THEN JJ$ = RIGHT$ (JJ$,1)
6360    IF LEN (JJ$) = 3 THEN JJ$ = RIGHT$ (JJ$,2)
6370  J = VAL (JJ$)
6380    IF J > NU THEN PRINT "NO SUCH VARIABLE. TRY AGAIN.":
          GOTO 6270
6390  A(KK,J) = JJ
6400    GOTO 6910
6410    IF LEN (JJ$) = 4 THEN JJ$ = MID$ (JJ$2,2)
6420    IF LEN (JJ$) = 3 THEN JJ$ = MID$ (JJ$,2,1)
6430  J = VAL (JJ$)
6440    IF J > MC THEN PRINT "NO SUCH DEV. VAR. TRY AGAIN.":
          GOTO 6270
6450  A(KK,J + NU + MC) = JJ
6460    GOTO 6910
6470    IF LEN (JJ$) = 4 THEN JJ$ = MID$ (JJ$,2,2)
6480    IF LEN (JJ$) = 3 THEN JJ$ = MID$ (JJ$2,1)
6490  J = VAL (JJ$)
6500    IF J > MC THEN PRINT "NO SUCH DEV.VAR. TRY AGAIN.":
          GOTO 6270
6510  A(KK,J + NU) + JJ
6520    GOTO 6910
6530    INPUT "WHAT DEV. VAR.'S PRIORITY? ";JJ$
6540    PRINT "WHAT IS NEW PRIORITY FOR "JJ$";: INPUT JJ
6550    IF LEFT$ (JJ$,1) < > "D" THEN PRINT "NEED A D TO
          PROCESS.": GOTO 6530
6560    IF RIGHT$ (JJ$,1) = "+" THEN GOTO 6580
6570    IF RIGHT$ (JJ$,1) = "-" THEN GOTO 6630
6580    IF LEN (JJ$) = 4 THEN JJ$ = MID$ (JJ HGR ,2,2)
6590    IF LEN (JJ$) = 3 THEN JJ$ = MID$ (JJ$,2,1)
6600  J = VAL (JJ$)
6610  C(J + NU + MC) = JJ
6620    GOTO 6670
6630    IF LEN (JJ$) = 4 THEN JJ$ = MID$ (JJ$,2,2)
6640    IF LEN (JJ$) = 3 THEN JJ$ = MID$ (JJ$,2,1)
6650  J = VAL (JJ$)
6660  C(J + NU) = JJ
6670    PRINT
6680    INPUT "ANOTHER PRIORITY? ";Q$
6690    IF Q$ = "Y" THEN GOTO 6530
6700    IF Q$ = "N" THEN GOTO 7030
6710    PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 6680
6720    INPUT "WHAT DEV. VAR.'S WEIGHT? ";JJ$
```

```
6730    PRINT "WHAT IS NEW WEIGHT FOR ";JJ$;: INPUT JJ
6740    IF LEFT$ (JJ$,1) < > "D" THEN GOTO 6770
6750    IF RIGHT$ (JJ$,1) = "+" THEN GOTO 6770
6760    IF RIGHT$ (JJ$,1) = "-" THEN GOTO 6820
6770    IF LEN (JJ$) = 4 THEN JJ$ = MID$ (JJ$,2,2)
6780    IF LEN (JJ$) = 3 THEN JJ$ = MID$ (JJ$,2,1)
6790  J = VAL (JJ$)
6800  WC(J + NU + MC) = JJ
6810    GOTO 6860
6820    IF LEN (JJ$) = 4 THEN JJ$ = MID$ (JJ$,2,2)
6830    IF LEN (JJ$) = 3 THEN JJ$ = MID$ (JJ$,2,1)
6840  J = VAL (JJ$)
6850  WC(J + NU) = JJ
6860    PRINT
6870    INPUT "ANOTHER WEIGHT? ";Q$
6880    IF Q$ = "Y" THEN GOTO 6720
6890    IF Q$ = "N" THEN GOTO 7100
6900    PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 6870
6910    PRINT
6920    INPUT "ANOTHER VALUE, SAME EQUATION? ";Q$
6930    IF Q$ = "Y" THEN GOTO 6270
6940    IF Q$ = "N" THEN GOTO 6960
6950    PRINT "Y OR N ONLY. TRY AGAIN.": GOTO 6920
6960    PRINT D$"OPEN";PR$;",L300"
6970    PRINT D$"WRITE";PR$;",R";KK
6980    FOR I = 1 TO NU + 2 * MC + 1
6990    PRINT A(KK,I) .
7000    NEXT I
7010    PRINT D$"CLOSE";PR$
7020    GOTO 7160
7030    PRINT D$"OPEN";PR$;",L300"
7040    PRINT D$"WRITE";PR$;",R";KK
7050    FOR I = 1 TO NU + 2 * MC
7060    PRINT C(I)
7070    NEXT I
7080    PRINT D$"CLOSE";PR$
7090    GOTO 7160
7100    PRINT D$"OPEN";PR$;",L300"
7110    PRINT D$"WRITE";PR$;",R";KK
7120    FOR I = 1 TO NU + 2 * MC
7130    PRINT WC(I)
7140    NEXT I
7150    PRINT D$"CLOSE";PR$
7160    PRINT
7170    INPUT "ANY MORE CHANGES?";Q$
7180    IF Q$ = "Y" THEN GOTO 6160
7190    IF Q$ = "N" THEN RETURN
7200    PRINT "Y OR N ONLY. TRY AGAIN": GOTO 7170
```

Appendix C - Computer Printout

ACSC14

D1- 4	WT 1
D2- 4	WT 1
D3- 4	WT 1
D4- 4	WT 1
D5- 4	WT 1
D6- 4	WT 1
D7- 4	WT 1
D8- 4	WT 1
D9- 4	WT 1
D10- 4	WT 1
D11- 4	WT 1
D12- 4	WT 1
D13- 4	WT 1
D14- 4	WT 1
D15- 4	WT 1
D1+ 4	WT 1
D2+ 2	WT 1
D3+ 4	WT 1
D4+ 3	WT 1
D5+ 1	WT 1
D6+ 4	WT 1
D7+ 4	WT 1
D8+ 4	WT 1
D9+ 4	WT 1
D10+ 4	WT 1
D11+ 4	WT 1
D12+ 4	WT 1
D13+ 4	WT 1
D14+ 4	WT 1
D15+ 4	WT 1

COEFFICIENTS IN TABLEAU:

D1- 10 3 5 1 2 9 2 3 1 1 5 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0
0 0 0 0
D2- 10 3 4 2 6 9 5 3 2 2 4 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0
0 0 0 0
(54)
D3- 10 4 3 4 1 2 3 3 4 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0
0 0 0 0
D4- 10 9 7 8 1 9 4 8 9 2 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0

```
D5- 10 3 3 1 1 9 1 3 8 1 3 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0
0 0 0 0
D6- 100 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
D7- 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
D8- 5 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
D9- 10 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D10- 5 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D11- 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D12- 8 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D13- 5 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D14- 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D15- 2 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0

VALUES IN ZJ-CJ:

P4 191 23 23 17 12 39 16 21 24 8 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -2 -1 -2 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
P3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
0 0 0
P2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0
0 0 0
P1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0
0 0 0
PIVOT COLUMN = 5
PIVOT ROW = 1
                                        55

COEFFICIENTS IN TABLEAU:

X5 1.11111111 .333333333 .555555556 .111111111 .222222222 1 .222222222 .33333333
3 .111111111 .111111111 .555555556 .111111111 0 0 0 0 0 0 0 0 0 0 0 0 0 -.1111
11111 0 0 0 0 0 0 0 0 0 0 0 0 0 0
D2- 0 0 -1 1 4 0 3 0 1 1 -1 -1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -1 0 0 0 0 0 0 0 0
0 0 0 0 0
D3- 7.77777778 3.33333333 1.88888889 3.77777778 .555555556 0 2.55555556 2.333333
33 3.77777778 .777777778 -.111111111 -.222222222 0 1 0 0 0 0 0 0 0 0 0 0 0 .22
2222222 0 -1 0 0 0 0 0 0 0 0 0 0 0 0
D4- 0 6 2 7 -1 0 2 5 7 1 -4 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0
0 0 0 0 0
D5- 0 0 -2 0 -1 0 -1 0 7 0 -2 -1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 -1 0 0 0 0
0 0 0 0 0 0
D6- 100 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
D7- 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
D8- 5 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
D9- 10 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0                                  (55)
D10- 3.88888889 -.333333333 -.555555556 -.111111111 -.222222222 0 -.222222222 -.
333333333 -.111111111 -.111111111 -.555555556 -.111111111 0 0 0 0 0 0 0 0 1 0 0
0 0 0 .111111111 0 0 0 0 0 0 0 0 0 0 0 0 0
D11- 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
D12- 8 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D13- 5 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

 0 0 0
D14- 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D15- 2 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0

VALUES IN ZJ-CJ:

P4 147.666667 10 1.33333334 12.6666667 3.33333334 0 7.33333334 8 19.6666667 3.66
666667 -6.66666667 -4.33333334 0 0 0 0 0 0 0 0 0 0 0 0 0 2.33333333 -1 -2 -1 -
1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
P3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
0 0 0
P2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0
0 0 0
P1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0
0 0 0
PIVOT COLUMN = 8
PIVOT ROW = 13
```

56

```
COEFFICIENTS IN TABLEAU:

X5 .555555556 .333333333 .555555556 .111111111 .222222222 1 .222222222 .33333333
3 0 .111111111 .555555556 .111111111 0 0 0 0 0 0 0 0 0 0 -.111111111 0 0 -.111
111111 0 0 0 0 0 0 0 0 0 0 0 0 0 0
D2- -5 0 -1 1 4 0 3 0 0 1 -1 -1 1 0 0 0 0 0 0 0 0 0 0 -1 0 0 1 -1 0 0 0 0 0 0 0
0 0 0 0 0 0
D3- -11.1111111 3.33333333 1.88888889 3.77777778 .555555556 0 2.55555556 2.33333
333 0 .777777778 -.111111111 -.222222222 0 1 0 0 0 0 0 0 0 0 0 -3.77777778 0 0 .
222222222 0 -1 0 0 0 0 0 0 0 0 0 0 0 0
D4- -35 6 2 7 -1 0 2 5 0 1 -4 -1 0 0 1 0 0 0 0 0 0 0 0 -7 0 0 1 0 0 -1 0 0 0 0 0
 0 0 0 0 0 0
D5- -35 0 -2 0 -1 0 -1 0 0 0 -2 -1 0 0 0 1 0 0 0 0 0 0 0 -7 0 0 1 0 0 0 -1 0 0 0
 0 0 0 0 0 0 0
D6- 100 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
D7- 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
D8- 5 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
D9- 10 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D10- 4.44444445 -.333333333 -.555555556 -.111111111 -.222222222 0 -.222222222 -.
333333333 0 -.111111111 -.555555556 -.111111111 0 0 0 0 0 0 0 0 1 0 0 .111111111
 0 0 .111111111 0 0 0 0 0 0 0 0 0 0 0 0 0
D11- 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D12- 8 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
X8 5 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0
D14- 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D15- 2 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0

VALUES IN ZJ-CJ:
```

(56)

```
P4 49.3333333 10 1.33333334 12.6666667 3.33333334 0 7.33333334 8 0 3.66666667 -6
 66666667 -4.33333334 0 0 0 0 0 0 0 0 0 0 -19.6666667 0 0 2.33333333 -1 -2 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
P3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
```

```
P2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0
0 0 0
P1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0
0 0 0
PIVOT COLUMN = 3
PIVOT ROW = 1

COEFFICIENTS IN TABLEAU:

X3 5 3 5 1 2 9 2 3 0 1 5 1 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 -1 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D2- -10 -3 -6 0 2 -9 1 -3 0 0 -6 -2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 -1 0 0 0 0 0 0
 0 0 0 0 0 0 0
D3- -30 -8 -17 0 -7 -34 -5 -9 0 -3 -19 -4 0 1 0 0 0 0 0 0 0 0 0 0 0 0 4 0 -1 0 0
 0 0 0 0 0 0 0 0 0 0
D4- -70 -15 -33 0 -15 -63 -12 -16 0 -6 -39 -8 0 0 1 0 0 0 0 0 0 0 0 0 0 0 8 0 0
-1 0 0 0 0 0 0 0 0 0 0 0
D5- -35 0 -2 0 -1 0 -1 0 0 0 -2 -1 0 0 0 1 0 0 0 0 0 0 0 -7 0 0 1 0 0 0 -1 0 0 0
 0 0 0 0 0 0 0
D6- 100 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
D7- 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
D8- 1.86264515E-09 -3 -5 0 -2 -9 -2 -3 0 -1 -5 -1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0
D9- 10 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D10- 5 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D11- 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D12- 8 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
X8 5 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0
D14- 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0
D15- 2 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0

VALUES IN ZJ-CJ:

P4 -14 -28 -62 0 -22 -114 -18 -30 0 -9 -70 -17 0 0 0 0 0 0 0 0 0 0 0 -7 0 0 15 -
1 -2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
P3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0
0 0 0
P2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0
0 0 0
P1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0
0 0 0

SOLUTION VARIABLES ARE:

X3                  5
D2-                 -10
D3-                 -30
D4-                 -70
D5-                 -35
D6-                 100
D7-                 2
D8-                 1.86264515E-09
D9-                 10
D10-                5                (57)
D11-                2
D12-                8
X8                  5
D14-                [illegible]
```

UNACHIEVED GOALS ARE:

P4 -14

END
DATED
FILM
8-88
DTIC